マイクロコンピュータZ80
ユーザーズマニュアル〔I〕

Microcomputer Z80
User's Manual〔Part I〕

THE SHARP MICROCOMPUTERS

ELECTRONICS DIGEST COMPANY

マイクロコンピュータZ80
ユーザーズマニュアル〔I〕

発行　シャープ株式会社
発売　株式会社エレクトロニクスダイジェスト書店部

Z-80-CPU テクニカルマニュアル 1

シャープZ－８０ マイクロコンピュータ[*1]は、マイクロコンピュータ・コンポーネント[*2]とその開発システム，サポート・ソフトウェア[*3]を完備し、容易にシステム設計ができるよう配慮されています。Z－８０ マイクロコンピュータ・コンポーネントを採用することにより、別の外部論理回路を付加しなくても高性能のマイクロコンピュータ・システムが得られ、最小限の低コスト標準メモリ[*4]を用いるだけでその目的が達成されます。

LH－００８０ Z－８０ CPU[*5]（以下、Z－８０ CPUと略します）は、Nチャネル・シリコンゲート E／D MOS プロセス[*6]で作られた、高度な処理能力を備えた第三世代のマイクロプロセッサ[*7]です。

Z－８０ CPUは、標準メモリの利用を考慮し、システムのスループット[*8]およびメモリの利用効率を高めた設計となっています。また周辺回路を制御するためのデコード[*9]された制御信号を持っています。Z－８０ CPUは＋５Vの単一電源および単相クロックを必要とするだけで、周辺回路が簡単になります。

1. 特　　長

- ８ビット並列処理のワンチップ・マイクロプロセッサ
- Nチャネル・シリコンゲート E／D MOS プロセス
- １５８種の基本命令（８０８０Aの７８種の全命令を含み、機械語においてソフトウェア互換性があります。８０８０Aに比べて、インデックス[*10]、ビットおよび相対アドレシング・モード[*11]や、４，８および１６ビット操作命令などの有効な命令が追加されています。）
- ２２のレジスタ[*12]内蔵
- 強力な割り込み機能[*13]：３モードのマスク可能な割り込み機能
  ：マスク不可能な割り込み機能
- 外部回路をほとんど必要としないで、標準的なスタティックやダイナミック・メモリ[*14]と直結可能（リフレッシュ回路[*15]内蔵）
- 命令フェッチ速度[*16]：１.６ μs
- ＋５Vの単一電源および単相クロック
- 全入出力端子：TTLコンパチブル[*17]
- パッケージは４０ピンDIP[*18]

2. 内　部　構　成

Z－８０ CPUの内部構成ブロック図を図１に示します。

---

* 1 microcomputer
* 2 microcomputer component
* 3 support software
* 4 memory
* 5 Central Processing Unit
* 6 N-channel silicon gate Enhancement／Depletion type Metal Oxide Semiconductor
* 7 microprocessor
* 8 system throughput
* 9 decode
*10 index
*11 relative addressing mode
*12 register
*13 interruption
*14 static or dynamic memory
*15 refresh circuit
*16 instruction fetch speed
*17 Transistor－Transistor Logic compatible
*18 Dual In line Package

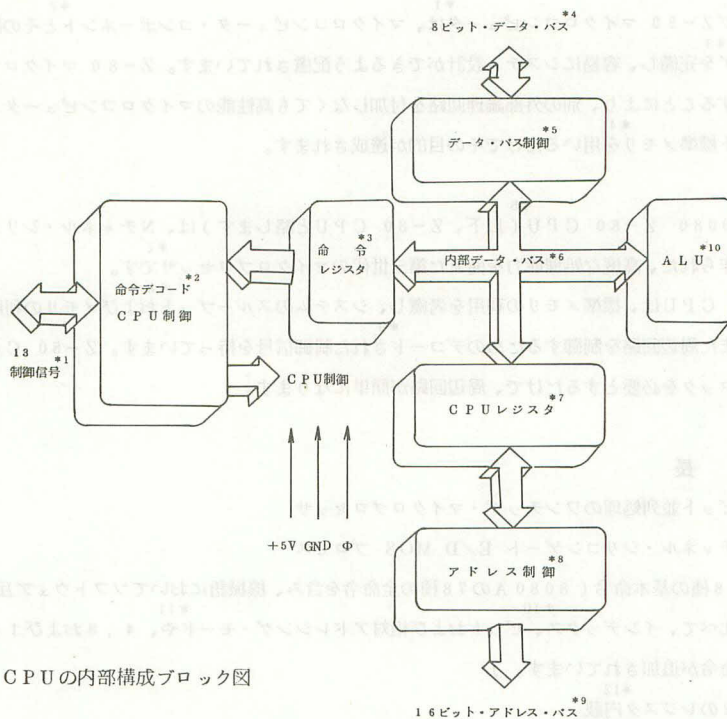



図1　Z－80　CPUの内部構成ブロック図

Z－80CPUの内部レジスタは、207ビットのリード／ライト・メモリ[*11]で構成されており、レジスタの構成を図2に示します。

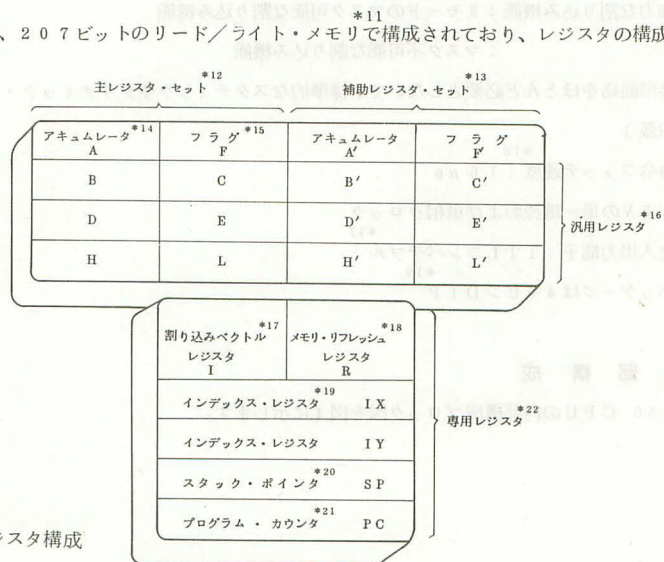



図2　Z－80CPUレジスタ構成

- ＊1　control signals
- ＊2　instruction decode and CPU control
- ＊3　instruction register
- ＊4　8－bit data bus
- ＊5　data bus control
- ＊6　internal data bus
- ＊7　CPU register
- ＊8　address control
- ＊9　16－bit address bus
- ＊10　arithmetic and logical unit
- ＊11　read／write memory
- ＊12　main register set
- ＊13　alternate register set
- ＊14　accumulator
- ＊15　flags
- ＊16　general purpose registers
- ＊17　interrupt vector register
- ＊18　memory refresh register
- ＊19　index register
- ＊20　stack pointer
- ＊21　program counter
- ＊22　special purpose registers

ＣＰＵレジスタは、汎用レジスタ群と専用レジスタ群から構成されており、汎用レジスタ群は、主レジスタ・セットと補助レジスタ・セットの２組のレジスタ・セットがあり、交換命令によって互いの内容を交換することができます。各レジスタ・セットは、８ビットのアキュムレータ、８ビットのフラグ・レジスタおよび６個の汎用レジスタ（各８ビット構成）で構成されています。汎用レジスタをＢＣ、ＤＥ、およびＨＬのようにペアにして、１６ビットのレジスタとしても使用できます。

専用レジスタ群は、割り込みベクトル・レジスタＩ（８ビット）、メモリ・リフレッシュ・レジスタＲ（７ビット）、２組のインデックス・レジスタＩＸ、ＩＹ（１６ビット）、スタック・ポインタＳＰ（１６ビット）およびプログラム・カウンタＰＣ（１６ビット）で構成されています。

割り込みベクトル・レジスタＩは、割り込み発生時に、割り込みサービス・ルーチン[＊1]の間接アドレスの上位８ビットを与え、下位８ビットは割り込みデバイスから与えられます。

メモリ・リフレッシュ・レジスタＲは、外部メモリとしてダイナミックＲＡＭを使用する場合のメモリ・リフレッシュ用のアドレスを自動的に発生します。

5. 端　子　信　号

Ｚ－８０ＣＰＵの端子信号を図３に示します。

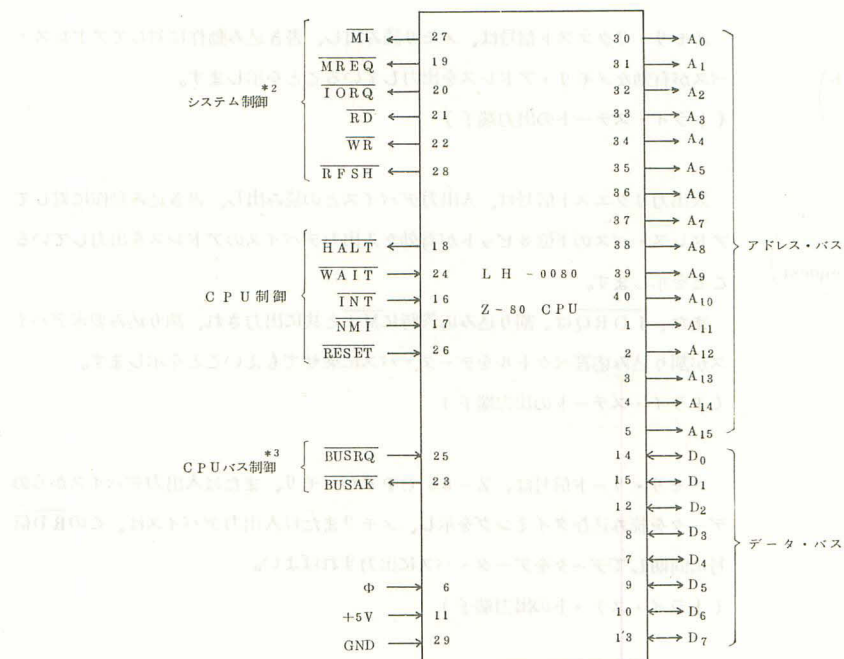



図３　Ｚ－８０ＣＰＵ端子信号

Ｚ－８０ＣＰＵの端子信号を機能的に分類しますと、

(1) アドレス・バス：A_0-A_{15}

(2) データ・バス　：D_0-D_7

＊1　interrupt service routine　　＊3　CPU bus control

＊2　system control

(3) システム制御　: $\overline{M_1}$, $\overline{MREQ}$, $\overline{IORQ}$, $\overline{WR}$, $\overline{RD}$, $\overline{RFSH}$

(4) ＣＰＵ制御　: $\overline{HALT}$, $\overline{WAIT}$, $\overline{INT}$, $\overline{NMI}$, $\overline{RESET}$

(5) ＣＰＵバス制御: $\overline{BUSRQ}$, $\overline{BUSAK}$

となります。次に各信号について説明します。

A_0-A_{15}
（アドレス・バス Address Bus）

A_0-A_{15}は、１６ビットのアドレス・バスを構成し、メモリ（最大６４Ｋバイト）および入出力デバイスのアドレスを指定します。
（トライ・ステート[*1]の出力端子）

D_0-D_7
（データ・バス Data Bus）

D_0-D_7は、８ビットの双方向性データ・バスを構成し、メモリおよび入出力デバイスとのデータの受渡しをおこないます。
（トライ・ステートの入出力端子）

$\overline{M1}$
（マシン・サイクル$_1$ Machine Cycle one）

$\overline{M1}$は、稼動中のマシン・サイクルがＯＰコードのフェッチ・サイクル[*2]であることを示します。
（出力端子）

$\overline{MREQ}$
（メモリ・リクエスト Memory Request）

メモリ・リクエスト信号は、メモリ読み出し、書き込み動作に対してアドレス・バスが有効なメモリ・アドレスを出力していることを示します。
（トライ・ステートの出力端子）

$\overline{IORQ}$
（入出力リクエスト Input/Output Request）

入出力リクエスト信号は、入出力デバイスとの読み出し、書き込み動作に対してアドレス・バスの下位８ビットが有効な入出力デバイスのアドレスを出力していることを示します。

また、$\overline{IORQ}$は、割り込み応答時に$\overline{M1}$と共に出力され、割り込み要求デバイスが割り込み応答ベクトルをデータ・バスに乗せてもよいことを示します。
（トライ・ステートの出力端子）

$\overline{RD}$
（メモリ・リード Memory Read）

メモリ・リード信号は、Ｚ－８０ ＣＰＵがメモリ、または入出力デバイスからのデータを読み込むタイミングを示し、メモリまたは入出力デバイスは、この$\overline{RD}$信号に同期してデータをデータ・バスに出力すればよい。
（トライ・ステートの出力端子）

$\overline{WR}$
（メモリ・ライト Memory Write）

メモリ・ライト信号は、アドレス指定されたメモリ、または入出力デバイスに書き込む有効データがデータ・バス上に乗っていることを示します。
（トライ・ステートの出力端子）

$\overline{RFSH}$
（リフレッシュ Refresh）

リフレッシュ信号は、ダイナミックＲＡＭ用のリフレッシュ用アドレスがアドレス・バスの下位７ビットに出力されていることを示します。このとき、$\overline{MREQ}$信号も出力されます。
（出力端子）

＊１　tri-state　　　　＊２　operation code fetch cycle

$\overline{\text{HALT}}$
(ホールト・ステート Halt State)

ホールト・ステート信号は、Z－80 CPUがHALT命令を実行中であることを示し、内部的にはNOP[*1]命令を実行しています。この間、メモリ・リフレッシュはおこなわれています。

ホールト状態の解除は、リセット信号、ノン・マスカブル割り込み[*2]、およびマスカブル割り込み[*3]（ただし、割り込み受け付け状態のとき）によっておこなわれます。

（出力端子）

$\overline{\text{WAIT}}$
(ウエイト Wait)

ウエイト信号は、アドレス指定されているメモリまたは、入出力デバイスがデータ転送準備のできていないことをZ－80 CPUへ知らせるための信号です。

この信号が入力されている間、Z－80 CPUは待ち状態を続けます。

（入力端子）

$\overline{\text{INT}}$
(割り込み要求 Interrupt Request)

入出力デバイスがZ－80 CPUに対して割り込みを要求する信号で、割り込み許可フリップ・フロップ[*4]がオン状態であれば、現在実行中の命令の終わりに、この割り込み要求が受け付けられます。

（入力端子）

$\overline{\text{NMI}}$
(ノン・マスカブル割り込み Non Maskable Interrupt)

このノン・マスカブル割り込みは、$\overline{\text{INT}}$より優先度の高い割り込み要求であり、ソフトウェアによってもマスクできません。$\overline{\text{NMI}}$はいつでも受け付けられて、現在実行中の命令が終わると割り込み処理が開始され、Z－80 CPUは自動的に0066_H番地から再スタートします。

（入力端子）

$\overline{\text{RESET}}$
(リセット Reset)

リセット信号は、割り込み許可フリップ・フロップ、プログラム・カウンタ、割り込みベクトル・レジスタ、およびメモリ・リフレッシュ・レジスタをリセットし、割り込みモードをモード0にして、Z－80 CPUを初期状態に戻します。

リセット期間には、アドレス・バスおよびデータ・バスは高インピーダンス状態となり、すべての制御信号も不活性状態となります。

（入力端子）

$\overline{\text{BUSRQ}}$
(バス・リクエスト Bus Request)

バス・リクエスト信号は、$\overline{\text{NMI}}$より優先度が高く、現在実行中のマシン・サイクルの終わりで受け付けられます。

この信号でZ－80 CPUのアドレス・バス、データ・バス、トライ・ステート制御出力信号が高インピーダンス状態になりますので、他のデバイスはこれらのバスを利用できるようになります。

（入力端子）

＊1　no operation instruction
＊2　non maskable interrupt
＊3　maskable interrupt
＊4　interrupt enable flip－flop

$\overline{\mathrm{BUSAK}}$
（バス・アクノリッジ / Bus Acknowledge）

バス・アクノリッジ信号は、Z－80 CPUがバス・リクエストを受け付けて、Z－80 CPUのアドレス・バス、データ・バスおよびトライ・ステート構造の制御出力のすべてが高インピーダンス状態になっていることを示します。この期間、外部デバイスはこれらのバスおよび制御線を利用できます。
（出力端子）

4. 主要タイミング波形

○ 命令OPコード・フェッチ・サイクル[*1]

命令サイクルの開始と同時に、プログラム・カウンタの内容がアドレス・バスへ出力され、これから半クロック遅れて$\overline{\mathrm{MREQ}}$が“L”になります。$\overline{\mathrm{MREQ}}$の立ち下がりエッジは、ダイナミック・メモリのチップ・イネイブル入力[*2]として直接使用することができます。$\overline{\mathrm{RD}}$信号によって、メモリのデータがデータ・バスへ出力され、Z－80 CPUはT_3の立ち上がりエッジでこのデータを読み込みます。フェッチ・サイクル（以下M1サイクルと呼びます）のT_3およびT_4は、ダイナミック・メモリのリフレッシュおよび命令の解読・実行が並行しておこなわれます。リフレッシュ制御信号$\overline{\mathrm{RFSH}}$によって、すべてのダイナミック・メモリのリフレッシュがおこなわれます。このときアドレス・バスにはリフレッシュ・アドレスが出力されています。タイミングを図4に示します。

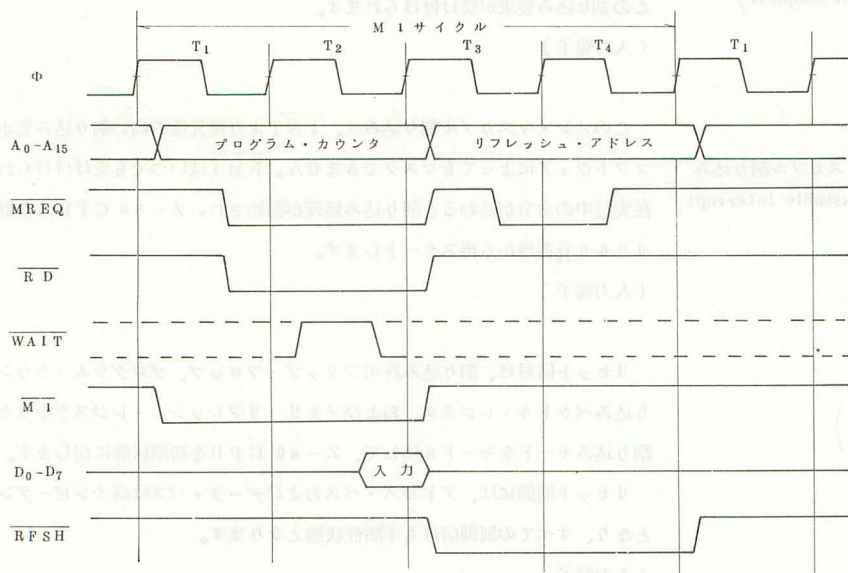



図4　命令OPコード・フェッチ・サイクル

○ メモリ・アクセス・タイミング[*3]

図5はM1サイクル以外のメモリ・アクセスのタイミングを示します。メモリ読み出しサイクルにおいて、$\overline{\mathrm{MREQ}}$および$\overline{\mathrm{RD}}$がM1サイクルとまったく同様に使用されます。$\overline{\mathrm{MREQ}}$はアドレス・バス上の情報が確定したとき“L”になるため、この信号はダイナミック・メモリのチップ・イネーブル入力として直接使用することができます。メモリ書き込みサイクルにおける$\overline{\mathrm{WR}}$は、データ・バス上の情報が確定したとき“L”になるため、一般のRAMのリード／ライトパルス[*4]として直接使用することができます。

＊1　instruction operation code fetch cycles
＊2　chip enable
＊3　memory access timing
＊4　read／write pulse

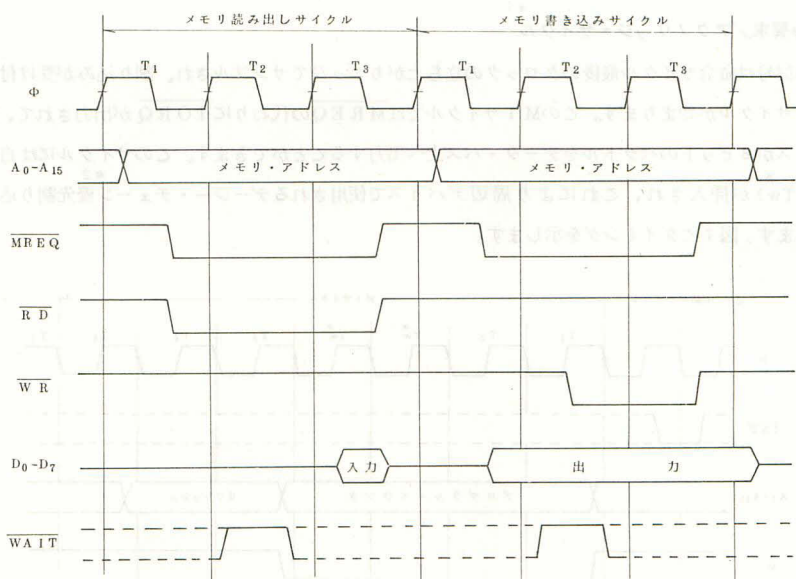



図5　メモリ・アクセス・タイミング

○ 入出力サイクル[*1]

図6に入出力サイクルのタイミングを示します。このサイクルでは、Z－80 CPUが自動的に1つの待ち状態（Tw*）を挿入し、入出力ポートがポート・アドレスをデコードする時間および必要ならば待ち状態の要求を出す時間を保証します。

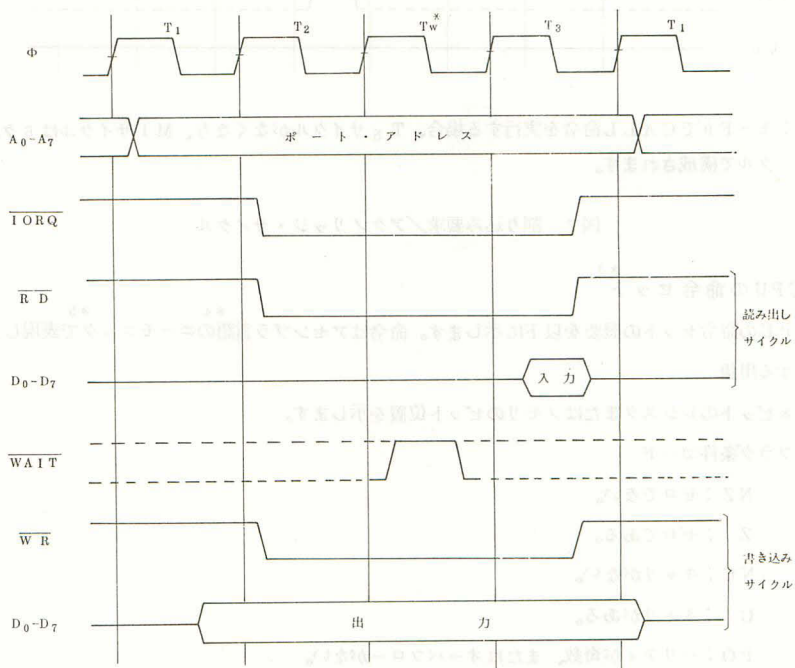



図6　入出力サイクル

＊1　input or output cycles

○ **割り込み要求／アクノリッジ・サイクル**[*1]

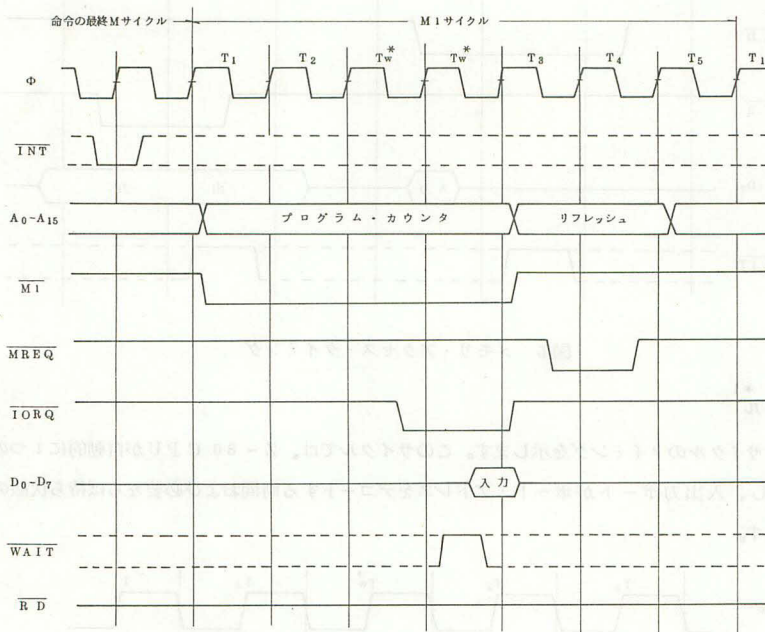
割り込み信号は命令サイクル最後のクロックの立ち上がりエッジでサンプルされ、割り込みが受け付けられると、特別なM1サイクルが始まります。このM1サイクルでは$\overline{MREQ}$の代わりに$\overline{IORQ}$が出力されて、割り込み要求したデバイスが8ビットのベクトルをデータ・バス上へ出力することができます。このサイクルには自動的に2つの待ち状態(Tw)[*]が挿入され、これにより周辺デバイスで使用されるデージー・チェーン[*2]優先割り込み機構を容易に実現できます。図7にタイミングを示します。

注：モード0でCALL命令を実行する場合、T_5サイクルがなくなり、M1サイクルは6クロック・サイクルで構成されます。

図7　割り込み要求／アクノリッジ・サイクル

5.　Z—80 CPUの命令セット[*3]

Z－80 CPUの命令セットの概要を以下に示します。命令はアセンブラ言語[*4]のニーモニック[*5]で表現します。

説明に使用する用語

b　　：8ビットのレジスタまたはメモリのビット位置を示します。

cc　：フラグ条件コード

NZ：ゼロでない。

Z　：ゼロである。

NC：キャリがない。

C　：キャリがある。

PO：パリティが奇数、またはオーバフローがない。

＊1　interupt request／acknowledge cycle
＊2　daisy chain
＊3　instruction set
＊4　assembly language
＊5　mnemonic

PE：パリティが偶数、またはオーバフローがある。

P　：符号が正である。

M　：符号が負である。

d　：命令を実行したときの8ビットのデータ格納場所を示します。

dd　：命令を実行したときの16ビットのデータ格納場所を示します。

e　：8ビットの符号付2進数で、相対アドレスまたはインデックス・アドレスを計算するとき使用されます。

L　：ゼロ・ページの特別なアドレスです。

n　：8ビットの2進数。

nn　：16ビットの2進数。

r　：8ビットの汎用レジスタを示します。（A，B，C，D，E，H，またはL）

s　：命令実行に使用する8ビットのデータのソースを示します。

Sb　：オペランド[*1]に示された8ビットのレジスタまたはメモリのビットを示します。

ss　：命令実行に使用する16ビットのデータのソースを示します。

添字L：16ビットのレジスタの下位8ビットを示します。

添字H：16ビットのレジスタの上位8ビットを示します。

(　)：(　)の内容はメモリまたは入出力ポートのポインタとして使用されます。

7ビットのレジスタはRです。

8ビットのレジスタはA，B，C，D，E，H，L，およびIです。

16ビットのレジスタ・ペア[*2]はAF，BC，DE，およびHLです。

16ビットのレジスタはSP，PC，IX，およびIYです。

アドレス方式は次に示す方式の組み合わせであってもよい。

イミディエット[*3]

拡張イミディエット[*4]

ゼロ・ページ修飾[*5]

相　対

直　接

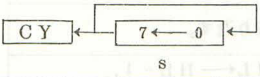
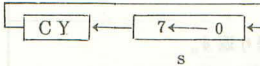
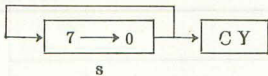
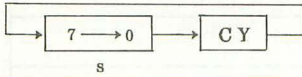
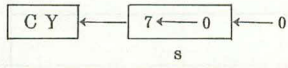
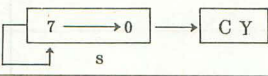
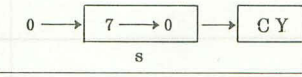
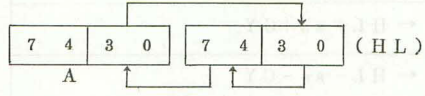
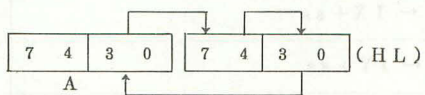
インデックス

レジスタ

インプライド[*6]

レジスタ間接[*7]

ビット

＊1　operand
＊2　register pair
＊3　immediate
＊4　immediate extended
＊5　modified page zero
＊6　implied
＊7　register indirect

	ニーモニック	動作内容	備考
8ビット・ロード	LD r, s	r ← s	s≡r, n, (HL), (IX+e), (IY+e)
	LD d, r	d ← r	d≡r, (HL), (IX+e), (IY+e)
	LD d, n	d ← n	d≡(HL), (IX+e), (IY+e)
	LD A, s	A ← s	s≡(BC), (DE), (nn), I, R
	LD d, A	d ← A	d≡(BC), (DE), (nn), I, R
16ビット・ロード	LD dd, nn	dd ← nn	dd≡BC, DE, HL, SP, IX, IY
	LD dd, (nn)	dd ← (nn)	
	LD (nn), ss	(nn) ← ss	ss≡BC, DE, HL, SP, IX, IY
	LD SP, ss	SP ← ss	ss≡HL, IX, IY
	PUSH ss	(SP−1) ← ss_H, (SP−2) ← ss_L	ss≡BC, DE, HL, AF, IX, IY
	POP dd	dd_L ← (SP), dd_H ← (SP+1)	dd≡BC, DE, HL, AF, IX, IY
交換	EX DE, HL	DE ⟷ HL	
	EX AF, AF′	AF ⟷ AF′	
	EXX	(BC, DE, HL) ⟷ (BC′, DE′, HL′)	
	EX (SP), ss	(SP) ⟷ ss_L, (SP+1) ⟷ ss_H	ss≡HL, IX, IY
メモリ・ブロック転送	LDI	(DE) ← (HL), DE ← DE+1 HL ← HL+1, BC ← BC−1	
	LDIR	LDIをBC＝0まで繰り返す。	
	LDD	(DE) ← (HL), DE ← DE−1 HL ← HL−1, BC ← BC−1	
	LDDR	LDDをBC＝0まで繰り返す。	
メモリ・ブロック・サーチ	CPI	A−(HL), HL ← HL+1 BC ← BC−1	A−(HL)はフラグを変えるだけで、Aの内容は不変である。
	CPIR	CPIをBC＝0またはA＝(HL)まで繰り返す。	
	CPD	A−(HL), HL ← HL−1 BC ← BC−1	
	CPDR	CPDをBC＝0またはA＝(HL)まで繰り返す。	
	ADD A, s	A ← A+s	CYはキャリ・フラグである。
	ADC A, s	A ← A+s+CY	

	ニーモニック	動　作　内　容	備　考
8ビット演算	SUB　s	A ← A－s	s≡r, n, (HL), (IX＋e), (IY＋e)
	SBC　A, s	A ← A－s－CY	
	AND　s	A ← A∧s	
	OR　s	A ← A∨s	
	XOR　s	A ← A⊕s	
	CP　s	A－s	Aの内容は不変。
	INC　d	d ← d＋1	d≡r, (HL), (IX＋e), (IY＋e)
	DEC　d	d ← d－1	
16ビット演算	ADD　HL, ss	HL ← HL＋ss	ss≡BC, DE, HL, SP
	ADC　HL, ss	HL ← HL＋ss＋CY	
	SBC　HL, ss	HL ← HL－ss－CY	
	ADD　IX, ss	IX ← IX＋ss	ss≡BC, DE, IX, SP
	ADD　IY, ss	IY ← IY＋ss	ss≡BC, DE, IY, SP
	INC　dd	dd ← dd＋1	dd≡BC, DE, HL, SP, IX, IY
	DEC　dd	dd ← dd－1	
アキュムレータ・フラグ操作	DAA	加減算後のAの内容の１０進補正を行う。	加減算はBCDの形でしておかねばならない。
	CPL	A ← $\overline{A}$	
	NEG	A ← 00_H －A	
	CCF	CY ← $\overline{CY}$	
	SCF	CY ← 1	
CPU制御	NOP	なにもしない。	
	HALT	CPUは停止する。	
	DI	割り込みディスエーブル。	
	EI	割り込みイネーブル。	
	IM　0	割り込みモード０にする。	8080Aと同じ
	IM　1	割り込みモード１にする。	0038_{16}からリスタートする。
	IM　2	割り込みモード２にする。	Iレジスタを使用して間接ジャンプを行う。
	RLC　s	CY ← [7 ← 0] s	s≡r, (HL), (IX＋e), (IY＋e)
	RL　s	CY ← [7 ← 0] s	

	ニーモニック	動　作　内　容	備　考
ローテーションおよびシフト	RRC　s	7 → 0 → CY（bit 0 → bit 7） s	
	RR　s	7 → 0 → CY → 7 s	
	SLA　s	CY ← 7 ← 0 ← 0 s	
	SRA　s	7 → 0 → CY（bit 7 保持） s	
	SRL　s	0 → 7 → 0 → CY s	s≡r，(HL)，(IX+e)，(IY+e)
	RLD	A：7 4 3 0　(HL)：7 4 3 0	
	RRD	A：7 4 3 0　(HL)：7 4 3 0	
ビット操作	BIT　b，s	Z ← $\overline{S}_b$	Zはゼロ・フラグである。
	SET　b，s	S_b ← 1	s≡r，(HL)，(IX+e)，(IY+e)
	RES　b，s	S_b ← 0	
リスタート	RST　L	(SP－1) ← PC_H，(SP－2) ← PC_L PC_H ← 0，PC_L ← L	
入出力	IN　A，(n)	A ← (n)	
	IN　r，(C)	r ← (C)	
	INI	(HL) ← (C)，HL ← HL+1， B ← B－1	
	INIR	INIをB=0まで繰り返す。	
	IND	(HL) ← (C)，HL ← HL－1， B ← B－1	
	INDR	INDをB=0まで繰り返す。	
	OUT　(n)，A	(n) ← A	
	OUT　(C)，r	(C) ← r	
	OUTI	(C) ← (HL)，HL ← HL+1， B ← B－1	
	OTIR	OUTIをB=0まで繰り返す。	
	OUTD	(C) ← (HL)，HL ← HL－1， B ← B－1	
	OTDR	OUTDをB=0まで繰り返す。	

	ニーモニック	動　作　内　容	備　考
ジャンプ	JP　nn	PC ← nn	
	JP　cc, nn	条件ccが成立するときPC ← nn, 成立しないとき続行。	cc { NZ PO Z PE NC P C M
	JR　e	PC ← PC+e	
	JR　kk, e	条件kkが成立するときPC ← PC+e, 成立しないとき続行。	kk { NZ NC Z C
	JP　(ss)	PC ← ss	ss≡HL, IX, IY
	DJNZ e	B ← B−1, B＝0のとき続行, B≠0のときPC ← PC+e	
コール	CALL　nn	(SP−1) ← PC_H, (SP−2)← PC_L PC ← nn	
	CALL　cc, nn	条件ccが成立するときCALL　nnと同じ、 成立しないとき続行。	cc { NZ PO Z PE NC P C M
リターン	RET	PC_L ← (SP), PC_H ← (SP+1)	
	RET　cc	条件ccが成立するときRETと同じ、成立しないとき続行。	cc { NZ PO Z PE NC P C M
	RETI	マスカブル割り込みからの復帰で、RETと同じ。	
	RETN	ノン・マスカブル割り込みからの復帰。	

6. 絶対最大定格

項　目	記号	定格値	単位
入力電圧	V_{IN}	−0.3〜+7	V
出力電圧	V_{OUT}	−0.3〜+7	V
動作温度	T_{opr}	0〜+70	℃
保存温度	T_{stg}	−65〜+150	℃

7. 電気的特性

7.1　DC特性

（Ta＝0℃〜+70℃, V_{CC} ＝+5V±5%）

記号	項　目	最小値	最大値	単位	測定条件
V_{ILC}	クロック“L”入力電圧	−0.3	0.45	V	
V_{IHC}	クロック“H”入力電圧	V_{CC}−0.6	V_{CC}+0.3	V	
V_{IL}	“L”入力電圧	−0.3	0.8	V	
V_{IH}	“H”入力電圧	2.0	V_{CC}	V	
V_{OL}	“L”出力電圧		0.4	V	I_{OL}＝1.8mA
V_{OH}	“H”出力電圧	2.4		V	I_{OH}＝−250μA
I_{CC}	消費電流		150	mA	
I_{LI}	入力リーク電流		10	μA	V_{IN}＝0〜V_{CC}
I_{LOH}	トライステート出力リーク電流		10	μA	V_{OUT}＝2.4V〜V_{CC}
I_{LOL}	トライステート出力リーク電流		−10	μA	V_{OUT}＝0.4V
I_{LD}	入力時のデータ・バスのリーク電流		±10	μA	0≦V_{IN}≦V_{CC}

7.2 端子容量

（Ta≒+25℃，f＝1MHz）

記号	項目	最大値	単位	測定条件
CΦ	クロック入力容量	50	pF	被測定端子以外の全ての端子は接地
C_{IN}	入力容量	8	pF	
C_{OUT}	出力容量	12	pF	

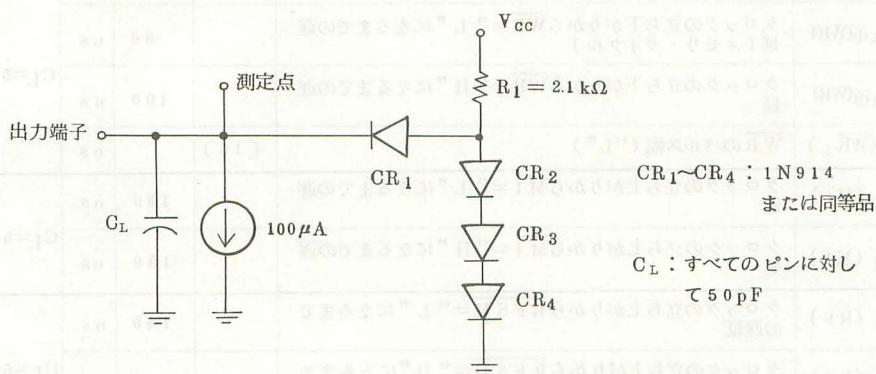
7.3 AC特性

（Ta＝0℃～+70℃，V_{CC}＝+5V±5％）

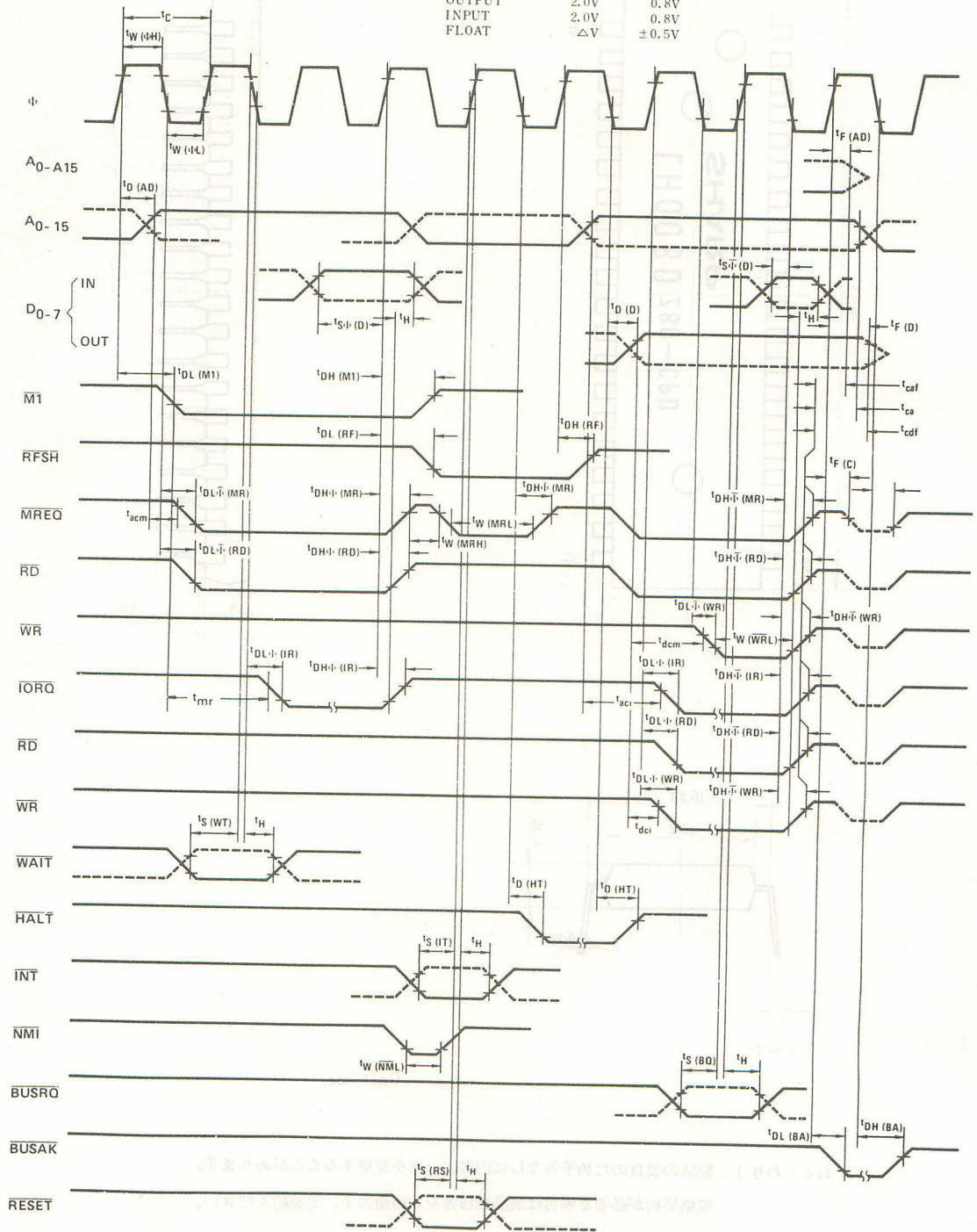
信号	記号	パラメータ	最小値	最大値	単位	測定条件
Φ	t_C	クロック周期	0.4	200	μs	
	$t_W(\Phi H)$	クロック・パルス幅（"H"）	180		ns	
	$t_W(\Phi L)$	クロック・パルス幅（"L"）	180	2000	ns	
	t_r，t_f	クロックの立ち上がり・立ち下がり時間		30	ns	
A_0-A_{15}	$t_D(AD)$	クロックの立ち上がりから出力までの遅延		145	ns	C_L=50pF
	$t_F(AD)$	出力がフロート状態になるまでの遅延		110	ns	
	t_{acm}	$\overline{MREQ}$に先立つ出力確定時間（メモリ・サイクル）	〔1〕		ns	
	t_{aci}	$\overline{IORQ}$，$\overline{RD}$または$\overline{WR}$に先立つ出力確定時間（入出力サイクル）	〔2〕		ns	
	t_{ca}	$\overline{RD}$，$\overline{WR}$，$\overline{IORQ}$または$\overline{MREQ}$からの出力保持時間	〔3〕		ns	
	t_{caf}	$\overline{RD}$または$\overline{WR}$からの出力保持時間（フロート状態への遷移時）	〔4〕		ns	
D_0-D_7	$t_D(D)$	クロックの立ち下がりから出力までの遅延		230	ns	C_L=50pF
	$t_F(D)$	出力がフロート状態になるまでの遅延（書き込みサイクル）		90	ns	
	$t_{S\Phi}(D)$	クロックの立ち上がりに対するセットアップ時間（M1サイクル）	50		ns	
	$t_{S\overline{\Phi}}(D)$	クロックの立ち下がりに対するセットアップ時間（M2～M5サイクル）	60		ns	
	t_{dcm}	$\overline{WR}$に先立つ出力確定時間（メモリ・サイクル）	〔5〕		ns	
	t_{dci}	$\overline{WR}$に先立つ出力確定時間（入出力サイクル）	〔6〕		ns	
	t_{cdf}	$\overline{WR}$からの出力保存時間	〔7〕		ns	
	t_H	ホールド時間	0		ns	
$\overline{MREQ}$	$t_{DL\overline{\Phi}}(MR)$	クロックの立ち下がりから$\overline{MREQ}$＝"L"になるまでの遅延		100	ns	C_L=50pF
	$t_{DH\Phi}(MR)$	クロックの立ち上がりから$\overline{MREQ}$＝"H"になるまでの遅延（M1サイクル）		100	ns	
	$t_{DH\overline{\Phi}}(MR)$	クロックの立ち下がりから$\overline{MREQ}$＝"H"になるまでの遅延（M2～M5サイクル）		100	ns	
	$t_W(\overline{MR_L})$	$\overline{MREQ}$のパルス幅（"L"）	〔8〕		ns	
	$t_W(\overline{MR_H})$	$\overline{MREQ}$のパルス幅（"H"）	〔9〕		ns	
$\overline{IORQ}$	$t_{DL\Phi}(IR)$	クロックの立ち上がりから$\overline{IORQ}$＝"L"になるまでの遅延（入出力サイクル）		90	ns	

信号	記号	パラメータ	最小値	最大値	単位	測定条件
$\overline{\text{IORQ}}$	$t_{DL}\overline{\Phi}(IR)$	クロックの立ち下がりから$\overline{\text{IORQ}}$="L"になるまでの遅延(INTAサイクル)		110	ns	C_L=50pF
	$t_{DH}\Phi(IR)$	クロックの立ち上がりから$\overline{\text{IORQ}}$="H"になるまでの遅延(INTAサイクル)		100	ns	
	$t_{DH}\overline{\Phi}(IR)$	クロックの立ち下がりから$\overline{\text{IORQ}}$="H"になるまでの遅延(入出力サイクル)		110	ns	
$\overline{\text{RD}}$	$t_{DL}\Phi(RD)$	クロックの立ち上がりから$\overline{\text{RD}}$="L"になるまでの遅延(入出力サイクル)		100	ns	C_L=50pF
	$t_{DL}\overline{\Phi}(RD)$	クロックの立ち下がりから$\overline{\text{RD}}$="L"になるまでの遅延(メモリ・サイクル)		130	ns	
	$t_{DH}\Phi(RD)$	クロックの立ち上がりから$\overline{\text{RD}}$="H"になるまでの遅延(M1サイクル)		100	ns	
	$t_{DH}\overline{\Phi}(RD)$	クロックの立ち下がりから$\overline{\text{RD}}$="H"になるまでの遅延(M2～M5サイクル)		110	ns	
$\overline{\text{WR}}$	$t_{DL}\Phi(WR)$	クロックの立ち上がりから$\overline{\text{WR}}$="L"になるまでの遅延(入出力サイクル)		80	ns	C_L=50pF
	$t_{DL}\overline{\Phi}(WR)$	クロックの立ち下がりから$\overline{\text{WR}}$="L"になるまでの遅延(メモリ・サイクル)		90	ns	
	$t_{DH}\overline{\Phi}(WR)$	クロックの立ち下がりから$\overline{\text{WR}}$="H"になるまでの遅延		100	ns	
	$t_W(\overline{WR_L})$	$\overline{\text{WR}}$のパルス幅("L")	〔10〕		ns	
$\overline{\text{M1}}$	$t_{DL}(M1)$	クロックの立ち上がりから$\overline{\text{M1}}$="L"になるまでの遅延		130	ns	C_L=50pF
	$t_{DH}(M1)$	クロックの立ち上がりから$\overline{\text{M1}}$="H"になるまでの遅延		130	ns	
$\overline{\text{RFSH}}$	$t_{DL}(RF)$	クロックの立ち上がりから$\overline{\text{RFSH}}$="L"になるまでの遅延		180	ns	C_L=50pF
	$t_{DH}(RF)$	クロックの立ち上がりから$\overline{\text{RFSH}}$="H"になるまでの遅延		150	ns	
$\overline{\text{WAIT}}$	$t_S(WT)$	クロックの立ち下がりに対するセットアップ時間	70		ns	
$\overline{\text{HALT}}$	t_D HT	クロックの立ち下がりからの遅延		300	ns	C_L=50pF
$\overline{\text{INT}}$	$t_S(IT)$	クロックの立ち上がりに対するセットアップ時間	80		ns	
$\overline{\text{NMI}}$	$t_W(\overline{NM_L})$	$\overline{\text{NMI}}$のパルス幅("L")	80		ns	
$\overline{\text{BUSRQ}}$	$t_S(BQ)$	クロックの立ち上がりに対するセットアップ時間	80		ns	
$\overline{\text{BUSAK}}$	$t_{DL}(BA)$	クロックの立ち上がりから$\overline{\text{BUSAK}}$="L"になるまでの遅延		120	ns	C_L=50pF
	$t_{DH}(BA)$	クロックの立ち下がりから$\overline{\text{BUSAK}}$="H"になるまでの遅延		110	ns	
$\overline{\text{RESET}}$	$t_S(RS)$	クロックの立ち上がりに対するセットアップ時間	90		ns	
	$t_F(C)$	フロート状態になるまでの遅延($\overline{\text{MREQ}}$, $\overline{\text{IORQ}}$, $\overline{\text{RD}}$および$\overline{\text{WR}}$)		100	ns	
	t_{mr}	$\overline{\text{IORQ}}$に先立つ$\overline{\text{M1}}$出力("L")の確定時間(INTAサイクル)	〔11〕		ns	

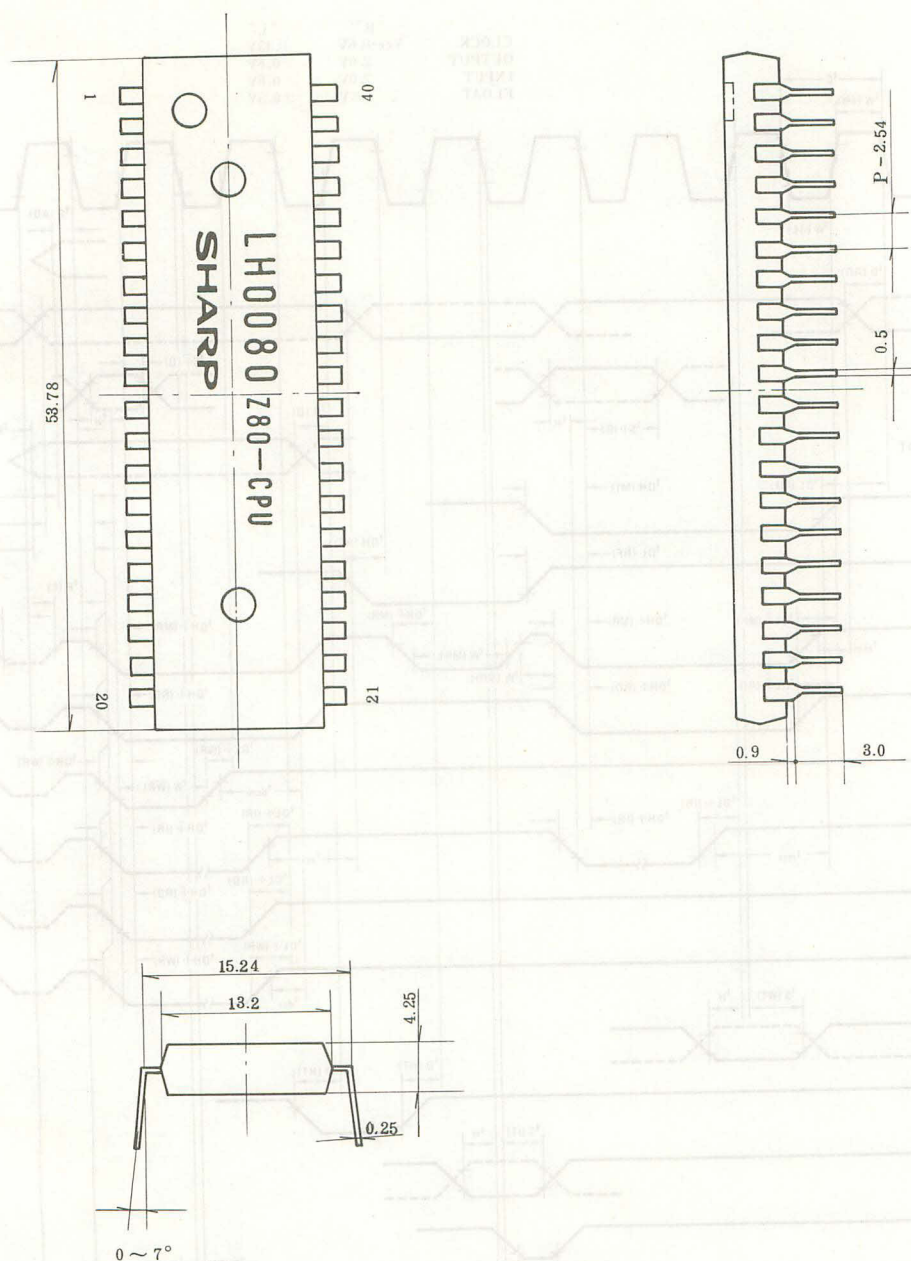
注
〔1〕 $t_{acm}=t_W(\Phi_H)+t_f-75$
〔2〕 $t_{aci}=t_C-80$
〔3〕 $t_{ca}=t_W(\Phi_L)+t_r-40$
〔4〕 $t_{caf}=t_W(\Phi_L)+t_r-60$
〔5〕 $t_{dcm}=t_C-210$
〔6〕 $t_{dci}=t_W(\Phi_L)+t_r-210$
〔7〕 $t_{cdf}=t_W(\Phi_L)+t_r-80$
〔8〕 $t_W(\overline{MR_L})=t_C-40$
〔9〕 $t_W(\overline{MR_H})=t_W(\Phi_H)+t_f-30$
〔10〕 $t_W(\overline{WR_L})=t_C-40$
〔11〕 $t_{mr}=2t_C+t_W(\Phi_H)+t_f-80$

- ○ データを$\overline{RD}$に同期してバスに送り出すことが望ましい。割り込みアクノリッジ・サイクルでは$\overline{M1}$をよび$\overline{IORQ}$の両方に同期して送り出すことが望ましい。
- ○ 制御信号はすべて内部で同期がとれているため、クロックについて非同期的に使用してもよい。
- ○ Ｔａ＝＋７０℃, V_{CC}＝＋５Ｖ±５％における負荷容量と出力の遅延との関係は次の通りです。
  負荷容量の５０ｐＦ増加につき遅延は１０ｎｓ増加します。負荷容量の最大値は、データ・バスが２００ｐＦで、他は１００ｐＦです。
- ○ $\overline{RESET}$の入力幅は最低３クロック・サイクル必要です。

出力端子測定回路

7.4　ＡＣタイミング図

測定条件は次の通りです。

	“H”	“L”
CLOCK	Vcc-0.6V	0.45V
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	△V	±0.5V

8. 外形寸法図

Unit：mm

（おことわり）　製品の改良のため予告なしに内容の一部を変更することがあります。

規格契約が必要な場合は製品仕様書をご用命の上、ご契約ください。

Z-80-PIO テクニカルマニュアル 2

シャープZ－８０マイクロコンピュータ[*1]は、マイクロコンピュータ・コンポーネント[*2]とその開発システム，サポート・ソフトウェア[*3]を完備し、容易にシステム設計ができるよう配慮されています。Z－８０マイクロコンピュータ・コンポーネントを採用することにより、別の外部論理回路を付加しなくても高性能のマイクロコンピュータ・システムが得られ、最小限の低コスト標準メモリ[*4]を用いるだけでその目的が達成されます。

LH－００８１　Z－８０　PIO（以下Z－８０　PIOと略します）は、Z－８０システムにおける、テープせん孔機、プリンタ、キーボードなどの周辺機器とZ－８０　CPUとの間の並列入出力インターフェース・コントローラ[*5]であり、プログラム可能な２ポート[*6]を持っています。

1. 特長

○　ハンドシェーク・データ転送制御[*7]機能を持つ２つの独立した８ビット双方向性インターフェース・ポート[*8]

○　Nチャネル・シリコンゲート　E／D　MOSプロセス[*9]

○　４動作モードの選択可能

　　バイト出力モード[*10]

　　バイト入力モード[*11]

　　バイト双方向性バス・モード[*12]　（ポートAのみ可能）

　　ビット・モード[*13]

○　周辺機器の状態によるプログラム可能な割り込み[*14]

○　自動的に割り込みベクタリング[*15]を行うデージー・チェーン[*16]による優先割り込み[*17]機能

○　ポートB出力は、ダーリントン・トランジスタ[*18]駆動可能

○　全入出力はTTLコンパチブル[*19]

○　＋５Vの単一電源、および単相クロック[*20]

○　パッケージは４０ピンDIP[*21]

＊1　microcomputer

＊2　microcomputer component

＊3　support software

＊4　memory

＊5　Parallel Input／Output interface controller

＊6　port

＊7　handshake data transfer control

＊8　bidirectional interface port

＊9　N－channel silicon gate Enhancement／Depletion type Metal Oxide Semiconductor

＊10　byte output mode

＊11　byte input mode

＊12　byte bidirectional bus mode

＊13　bit mode

＊14　programmable interrupt

＊15　interrupt vectoring

＊16　daisy chain

＊17　priority interrupt

＊18　darlington transistor

＊19　Transistor－Transistor Logic compatible

＊20　single phase clock

＊21　Dual In line Package

2. 内部構成

Z－８０　PＩＯのブロック構成図を図１に示します。Z－８０　PＩＯは、CPUバス入出力回路[*1]、内部制御回路、ポートAの入出力回路、ポートBの入出力回路および割り込み制御回路から構成されています。

Z－８０　PＩＯの典型的な使用法は、ポートAをデータ転送のチャネルとして使用し、ポートBをステータスおよび制御のモニタ[*2]として使用する方法です。

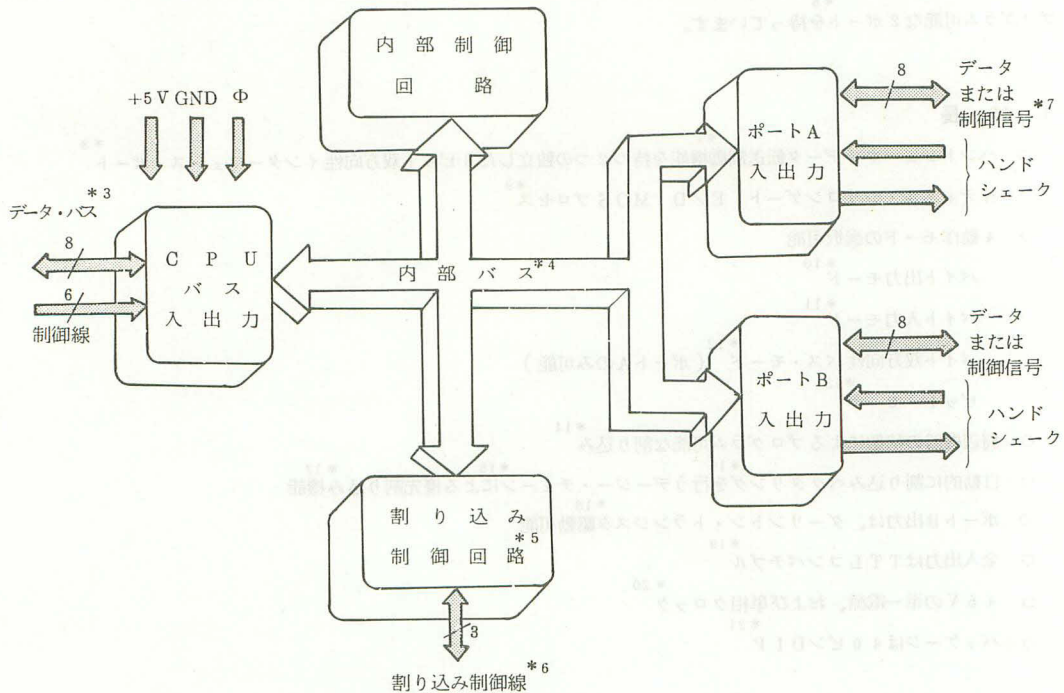



図１　Z－８０　PＩＯのブロック構成図

---

*1 CPU bus input／output logic

*2 status and control monitor

*3 data bus

*4 internal bus

*5 interrupt control logic

*6 interrupt control line

*7 data or control signal

ポートAおよびポートBの入出力部は、図2に示すようにハンドシェーク制御回路と6つのレジスタ[*1]から構成されています。

6つのレジスタとは、8ビットの入力レジスタ、8ビットの出力レジスタ、2ビットのモード制御レジスタ[*2]、8ビットのマスク・レジスタ[*3]、8ビットの入出力選択レジスタ[*4]、および2ビットのマスク制御レジスタ[*5]です。最後の3つのレジスタは、ポートがビット・モードのときにだけ使用されます。

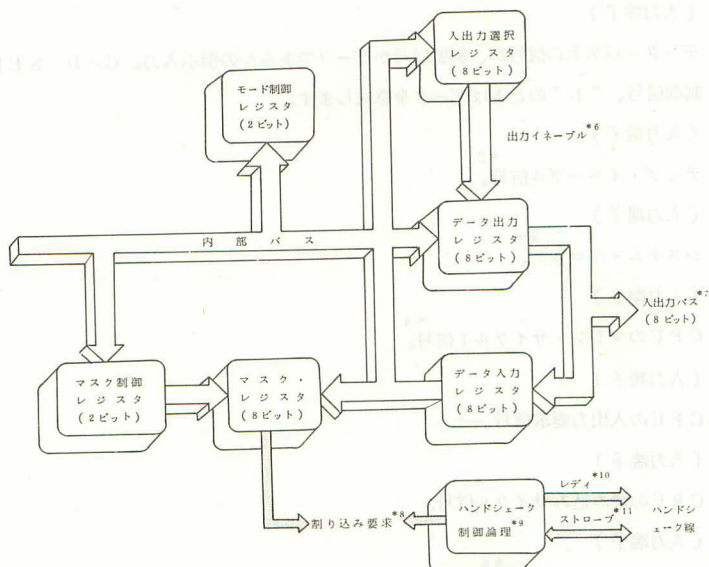



図2　各ポートの入出力ブロック構成図

次に6つのレジスタについて簡単に説明します。

モード制御レジスタ；2ビットのレジスタであり、CPUにより書き込まれて、バイト出力モード、バイト入力モード、バイト双方向性バス・モード、およびビット・モードの4動作モードの選択をおこないます。

データ出力レジスタ；8ビットのレジスタであり、CPUから周辺機器へ転送するデータを保持します。

データ入力レジスタ；8ビットのレジスタであり、周辺機器からCPUへ転送されるデータを受け取ります。

マスク制御レジスタ；2ビットのレジスタであり、モニタすべき周辺機器とのインターフェース端子の有効状態（1あるいは0）を定義し、マスクされていない全端子が有効状態（AND条件）のときか、それとも、少なくとも1つが有効状態（OR条件）のときのどちらの状態のときに割り込みを発生するかを示します。

マスク・レジスタ；8ビットのレジスタであり、マスク制御レジスタで示された条件にしたがって、周辺機器とのインターフェース用のどの端子をモニタすべきかを示します。

入出力選択レジスタ；8ビットのレジスタであり、ビット・モードにおいて、それぞれの端子を入力とするか出力とするかを指定します。

---

*1　register
*2　mode control register
*3　mask register
*4　input／output select register
*5　mask control register
*6　output enable
*7　input／output bus
*8　interrupt request
*9　handshake control logic
*10　ready
*11　strobe

3. 端子信号

$D_0 - D_7$　　CPUデータバス。
（トライ・ステート[*1]の入出力端子）

B／A　SEL　　ポートBとポートAの選択入力。B／A　SELが"H"のときはポートB、"L"のときはポートAを指定します。
（入力端子）

C／D　SEL　　データ・バス上の信号が、制御信号かデータであるかの指示入力。C／D　SELが"H"のときは制御信号、"L"のときはデータを意味します。
（入力端子）

$\overline{CE}$　　チップ・イネーブル信号[*2]。
（入力端子）

Φ　　システム・クロック[*3]。
（入力端子）

$\overline{M1}$　　CPUのマシン・サイクル1信号[*4]。
（入力端子）

$\overline{IORQ}$　　CPUの入出力要求信号。
（入力端子）

$\overline{RD}$　　CPUの読み込みサイクル信号[*5]。
（入力端子）

IEI　　割り込みイネーブル入力[*6]。IEIが"H"のとき、割り込み要求を出すとCPUに受け付けられます。
（入力端子）

IEO　　割り込みイネーブル出力[*7]。IEIとIEOは、優先割り込み機構に必要なデージー・チェーンを構成します。
（出力端子）

$\overline{INT}$　　CPUに対する割り込み要求信号。
（オープン・ドレイン[*8]，出力端子）

$A_0 - A_7$　　ポートAのバスライン。
（トライ・ステートの入出力端子）

$\overline{A\ STB}$　　周辺機器が与えるポートAのストローブ・パルス[*9]。
（入力端子）

A　RDY　　ポートAのデータ入出力レジスタがレディ状態であることを示します。
（出力端子）

$B_0 - B_7$　　ポートBのバスライン。
（トライ・ステートの入出力端子）

---

*1　tri-state
*2　chip enable
*3　system clock
*4　machine cycle one signal
*5　read cycle
*6　interrupt enable input
*7　interrupt enable output
*8　open drain
*9　strobe pulse

$\overline{\text{B STB}}$　周辺機器が与えるポートBのストローブ・パルス。
（入力端子）

B RDY　ポートBのデータ入出力レジスタがレディ状態であることを示します。
（出力端子）

Z－80　PIOの端子信号を下に示します。

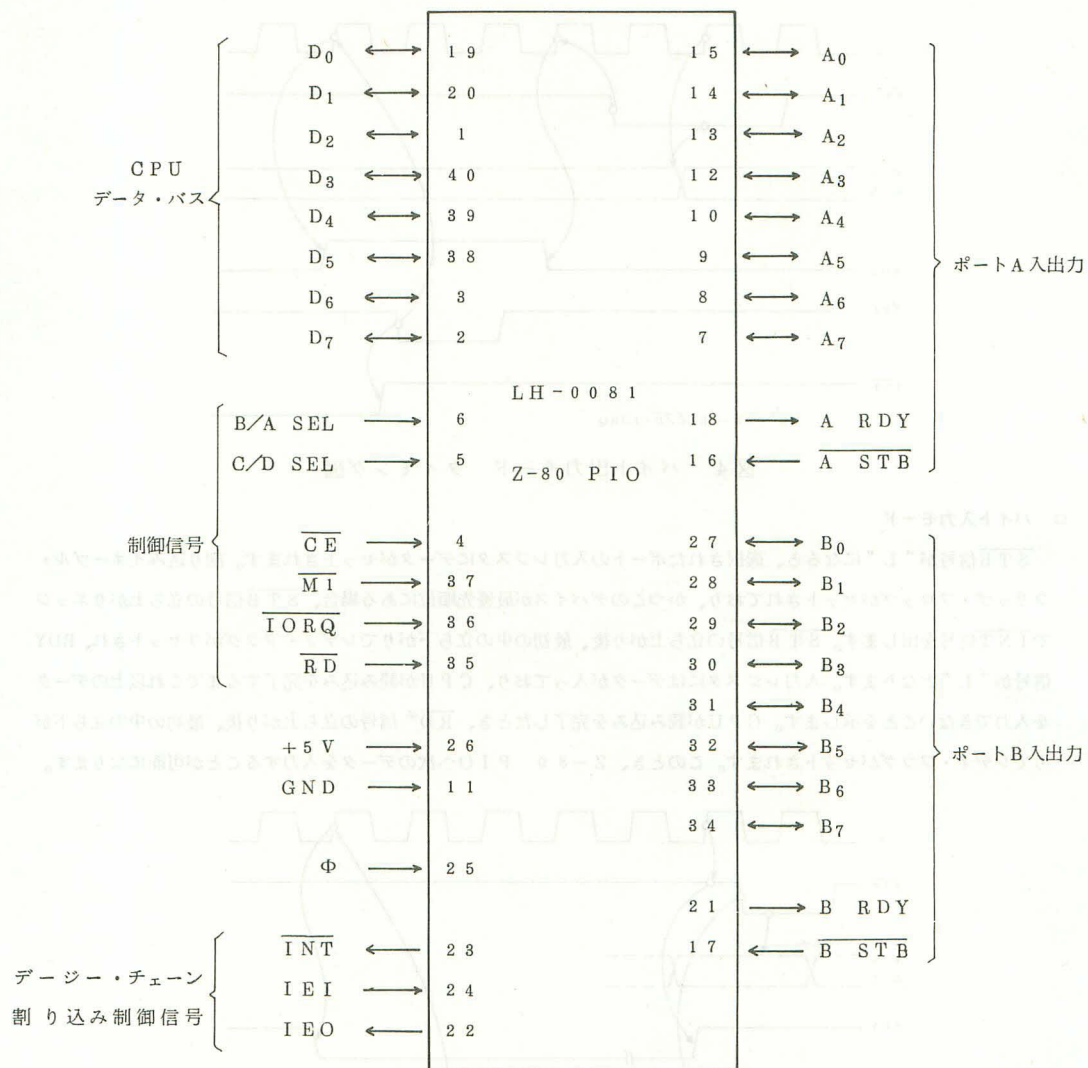



図3　Z－80　PIO端子信号

4. 主要タイミング波形

○　バイト出力モード

CPUが出力命令を実行することにより、出力サイクルが始まります。

$\overline{WR}^*$信号により、データ・バス上のデータが選択されたポートの出力レジスタにラッチ[*1]されます。$\overline{WR}^*$信号の立ち上がり後、最初のΦの立ち下がりでレディ・フラグ[*2]がセットされ、RDY信号が出力されて、データの利用できることを示します。周辺デバイスがデータを受け取ったことを示す$\overline{STB}$信号の立ち上がり後、最初のΦの立ち下がりでレディ・フラグがリセット[*3]されて、RDY信号が"L"となります。割り込みイネーブル・フリップ・フロップ[*4]がセットされており、かつこのデバイスが最優先順位にある場合、$\overline{STB}$信号の立ち上がりエッジ[*5]で$\overline{INT}$信号を発生して、CPUに対して割り込みを要求します。

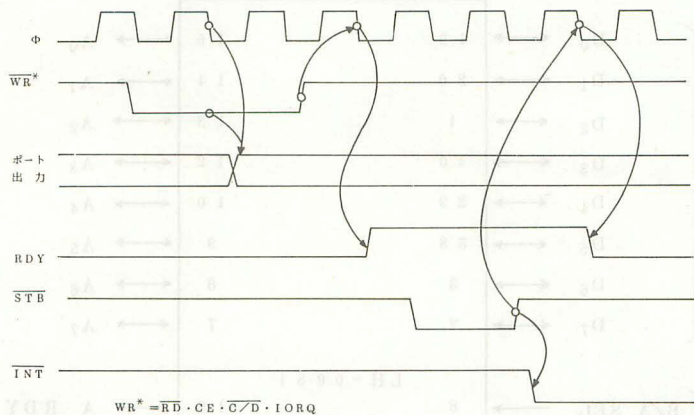



図4　バイト出力モード　タイミング図

○ **バイト入力モード**

$\overline{STB}$信号が"L"になると、選択されたポートの入力レジスタにデータがセットされます。割り込みイネーブル・フリップ・フロップがセットされており、かつこのデバイスが最優先順位にある場合、$\overline{STB}$信号の立ち上がりエッジで$\overline{INT}$信号を出します。$\overline{STB}$信号の立ち上がり後、最初のΦの立ち下がりでレディ・フラグがリセットされ、RDY信号が"L"となります。入力レジスタにはデータが入っており、CPUが読み込みを完了するまでこれ以上のデータを入力できないことを示します。CPUが読み込みを完了したとき、$\overline{RD}^*$信号の立ち上がり後、最初のΦの立ち下がりでレディ・フラグがセットされます。このとき、Z－80　PIOへ次のデータを入力することが可能になります。

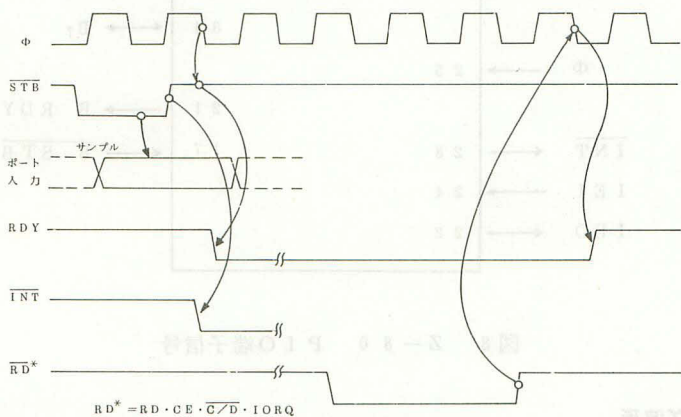



図5　バイト入力モード　タイミング図

---

*1 latch
*2 ready flag
*3 reset
*4 interrupt enable flip−flop
*5 edge

○　バイト双方向性モード

これは前記の出力モードと入力モードの組み合わせであり、4本のハンドシェーク線全部と8本のポートAバスを使用します。ポートBはビット・モードにしなければなりません。ポートAのハンドシェーク線は出力制御に使用され、ポートBのハンドシェーク線は入力制御に使用されます。$\overline{A\ \ STB}$が“L”であるときのみ、データがポートAバスに出力されます。このストローブの立ち上がりエッジにより、データを周辺機器にラッチすることができます。

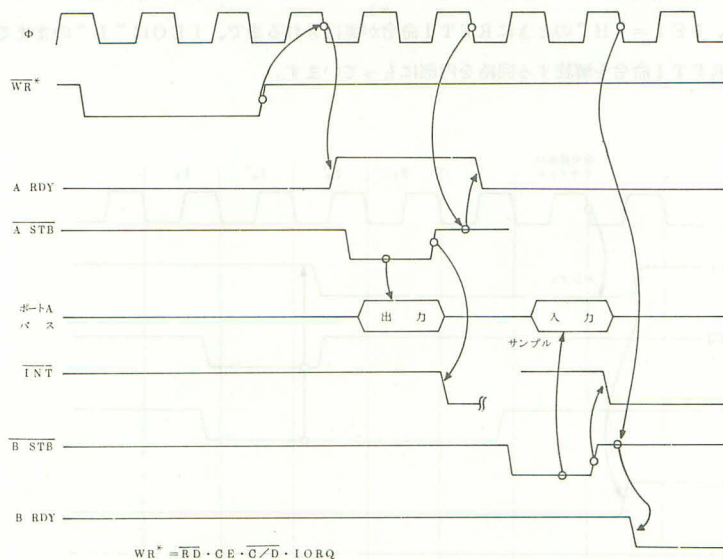



図6　バイト双方向性モード　タイミング図

○　ビット・モード

ビッド・モードではハンドシェーク信号を使用せず、通常のポートの読み出しおよび書き込みがいつでも実行できます。書き込みのときには、出力モードと同じタイミングでデータが出力レジスタにラッチされます。

読み出しのときに、CPUが受け取るデータは、出力に割り当てられたビットに対応する出力レジスタのビット・データおよび入力に割り当てられたビットに対応する入力レジスタのビット・データとで構成されます。入力レジスタは

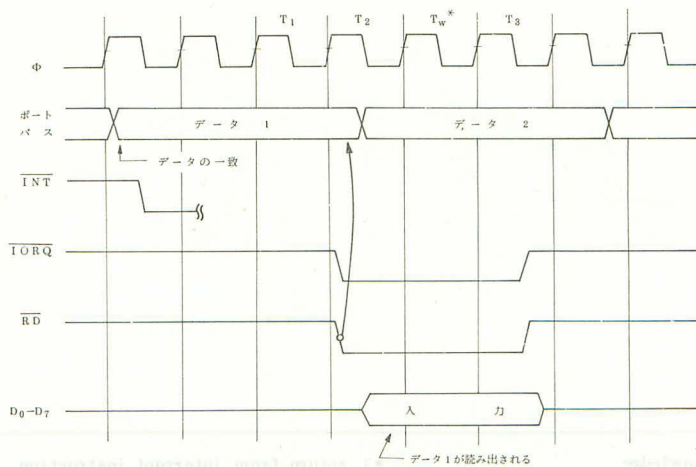



図7　ビット・モード　タイミング図

$\overline{RD}$の立ち下がり直前のデータを保持しています。割り込みイネーブル状態で、かつポート・データが８ビットのマスク・レジスタと２ビットのマスク制御レジスタで定義される論理条件を満足するとき、割り込みが発生します。

○ **割り込みアクノリッジ・サイクル**[*1]

$\overline{M1}$が“Ｌ”である間、割り込みイネーブル信号がデージー・チェーン中で確定することを保証するため、周辺コントローラ[*2]は割り込みイネーブルの状態を変更することが禁止されます。割り込みアクノリッジ・サイクルでＩＥＩ＝“Ｈ”かつＩＥＯ＝“Ｌ”である周辺デバイスが、前もってプログラムされている８ビットの割り込みベクトルをデータ・バス上へ送り出します。ＩＥＩ＝“Ｈ”のときにＲＥＴＩ命令[*3]が実行されるまで、ＩＥＯは“Ｌ”のままです。Ｚ－８０ ＰＩＯは２バイトのＲＥＴＩ命令を解読する回路を内部にもっています。

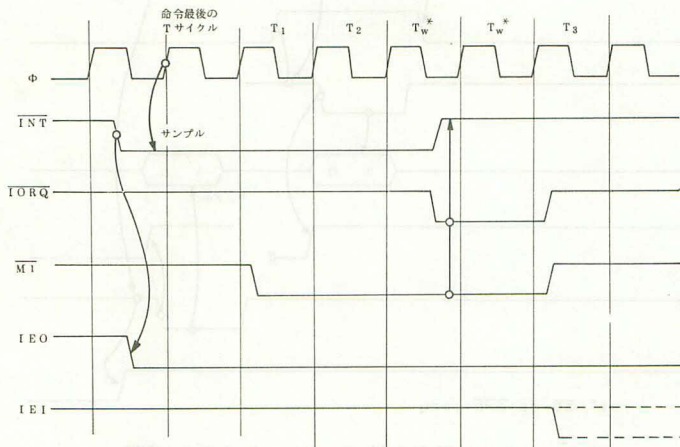



図８　割り込みアクノリッジ・サイクル

---

*1　interrupt acknowledge　　　*3　return from interrupt instruction

*2　peripheral controller

○　割り込みからの復帰サイクル

周辺デバイスが割り込み要求をしていない場合、または、割り込み処理がおこなわれていない場合、そのデバイスのＩＥＯとＩＥＩは等しい。

ＣＰＵが割り込み処理をおこなっている場合（すなわち、すでに割り込み要求を出し、かつ割り込みアクノリッジを受け取っている場合）、そのデバイスのＩＥＯは常に“Ｌ”であり、優先順位の低いデバイスからの割り込みを禁止します。もし、割り込み要求を出しても割り込みアクノリッジを受けていないデバイスがある場合、ＩＥＯは“Ｌ”になっていますが、２バイトＯＰコード*1の最初のバイトとして“ＥＤ”（１６進）がデコード*2されると、ＩＥＯは“Ｈ”になり、次のＯＰコードがデコードされると、ＩＥＩは再び“Ｌ”に戻ります。（ＣＰＵ内の割り込みイネーブル・フリップ・フロップが“Ｌ”になっているとき、この状態が発生します。）もし、２バイト目のＯＰコードが“４Ｄ”であれば、命令はＲＥＴＩです。

したがって、ＯＰコード“ＥＤ”がデコードされた後、現在割り込み処理を受けている周辺デバイスだけが、ＩＥＩ＝“Ｈ”かつＩＥＯ＝“Ｌ”という状態におかれます。このデバイスは、デージー・チェーン中の割り込みアクノリッジを受取った最優先順位のデバイスであり、他のすべての周辺デバイスはＩＥＩ＝ＩＥＯです。次にデコードされたＯＰコードが“４Ｄ”であると、この最優先順位のデバイスは、“割り込み処理中”という状態を解除します。

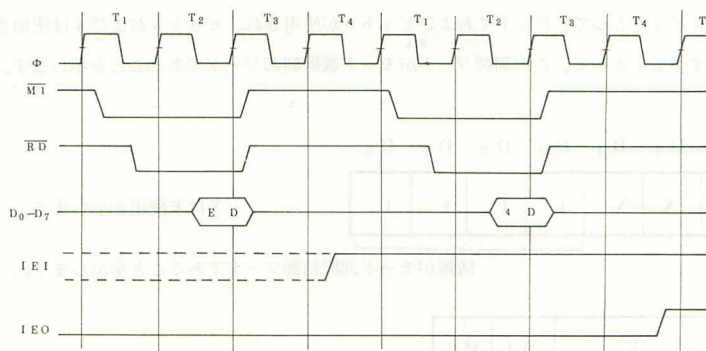



図９　割り込みからの復帰サイクル

＊1 operation code　　＊2 decode

5. 動作条件のプログラム

○ 割り込みベクトルの書き込み [*1]

割り込みをかけているデバイスは、ＣＰＵに８ビットの割り込みベクトルを与えることが必要であり、ＣＰＵはこのベクトルを使用して割り込みサービス・ルーチン [*2] の番地を作ります。割り込みアクノリッジ・サイクルにおいて、サービスを要求しているデバイスのうち最も優先順位の高いものがデータ・バス上へベクトルを送り出します。必要な割り込みベクトルは、C／D　SEL＝“Ｈ”として、ＣＰＵから各ポートの割り込みベクトル・レジスタへ次の形式で書き込まれます。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	0

この情報が割り込みベクトルであることを示します。

○ 動作モードの選択 [*3]

動作モードの選択は、２ビットのモード制御レジスタへデータを次の形式で書き込むことによりおこなわれます。動作選択のための２ビットとして、ビット７およびビット６が使用され、ビット５および４は使用されません。ビット３からビット０はすべて１として、この制御ワード [*4] がモード選択制御ワードであることを示します。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
M1	M0	X	X	1	1	1	1

Xは未使用を示します。

情報がモード選択制御ワードであることを示します。

モ　ー　ド	M1	M0
バイト出力モード	0	0
バイト入力モード	0	1
バイト双方向性バス・モード	1	0
ビット・モード	1	1

バイト出力モード　　データがＣＰＵから周辺デバイスへ転送されます。

バイト入力モード　　データが周辺デバイスからＣＰＵへ転送されます。

バイト双方向性バス・モード　　データがＣＰＵと周辺デバイスとの間で相互に授受されます。

ビット・モード　　このモードを選択したとき、各ポートのそれぞれのビットが入力か出力かを指示するための制御ワードを次に入出力選択レジスタへ書き込むことが必要です。

---

*1 load interrupt vector
*2 interrupt service routine
*3 selecting an operating mode
*4 control word

I/O＝1；入力

I/O＝0；出力

○ **割り込みの制御**

割り込み制御用の制御ワードの形式を次に示します。

ビット7＝1　　割り込みイネーブル・フリップ・フロップがセットされ、割り込みを発生できます。

ビット7＝0　　割り込みイネーブル・フリップ・フロップがリセットされて、割り込みを発生できません。

ビット6～4　　ビット・モードにおける割り込み条件を定めます。他のモードでは無視されます。

ビット3～0　　この情報が割り込み制御ワードであることを示します。

ビット4＝1のとき、次の制御ワードはマスク・レジスタに書き込むものでなければなりません。

D7	D6	D5	D4	D3	D2	D1	D0
MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0

MB＝0であるポート・データ線のみがモニタされ、割り込み条件が満たされれば割り込みを発生します。

ポートの割り込みイネーブル・フリップ・フロップは、割り込み制御ワードの代わりに、次の形式の情報でもセットあるいはリセットができます。

D7	D6	D5	D4	D3	D2	D1	D0
割り込みイネーブル	X	X	X	0	0	1	1

Z－80　PIOの入力端子として、リセット端子がなく、Z－80　PIOをリセットするには、$\overline{IORQ}\cdot\overline{RD}$＝“H”として$\overline{M1}$＝“L”を2クロック以上入力する必要があり、内部でリセット・フリップ・フロップがセットされ、Z－80　PIOのリセットがおこなわれます。このリセット・フリップ・フロップは、制御ワードの書き込みをおこなうと解除されます。

6. 絶対最大定格

項　目	記 号	定 格 値	単 位
入 力 電 圧	V_{IN}	−0.3～+7	V
出 力 電 圧	V_{OUT}	−0.3～+7	V
動 作 温 度	T_{opr}	0～+70	℃
保 存 温 度	T_{stg}	−65～+150	℃

7. 電気的特性

7.1 DC特性

（Ta＝0℃～+70℃，V_{CC}＝+5V±5%）

記 号	項　目	最小値	最大値	単位	測 定 条 件
V_{ILC}	クロック“L”入力電圧	−0.3	0.45	V	
V_{IHC}	クロック“H”入力電圧	V_{CC}−0.6	V_{CC}+0.3	V	
V_{IL}	“L”入力電圧	−0.3	0.8	V	
V_{IH}	“H”入力電圧	2.0	V_{CC}	V	
V_{OL}	“L”出力電圧		0.4	V	I_{OL}＝2mA
V_{OH}	“H”出力電圧	2.4		V	I_{OH}＝−250μA
I_{CC}	消 費 電 流		70	mA	
I_{LI}	入力リーク電流		10	μA	V_{IN}＝0～V_{CC}
I_{LOH}	トライステート出力リーク電流		10	μA	V_{OUT}＝2.4V～V_{CC}
I_{LOL}	トライステート出力リーク電流		−10	μA	V_{OUT}＝0.4V
I_{LD}	入力時のデータ・バスのリーク電流		±10	μA	0≦V_{IN}≦V_{CC}
I_{OHD}	ダーリントン駆動電流	−1.5		mA	V_{OH}＝1.5V ポートBのみ

7.2 端子容量

（Ta＝+25℃，f＝1MHz）

記 号	項　目	最大値	単 位	測 定 条 件
C_{Φ}	クロック入力容量	12	pF	被測定端子以外の全ての端子は接地
C_{IN}	入 力 容 量	7	pF	
C_{OUT}	出 力 容 量	10	pF	

7.3　ＡＣ特性　　　　　　　　　　　　（Ta＝０℃〜＋７０℃，Vcc＝＋５Ｖ±５％）

信　号	記　号	パ　ラ　メ　ー　タ	最小値	最大値	単位	測定条件
Φ	t_C	クロック周期	４００	〔１〕	ns	
	$t_W(\Phi H)$	クロック・パルス幅（"H"）	１７０	２０００	ns	
	$t_W(\Phi L)$	クロック・パルス幅（"L"）	１７０	２０００	ns	
	t_r，t_f	クロック立ち上がり・立ち下がり時間		３０	ns	
	t_H	ホールド時間	０		ns	
$\overline{CE}$，C/D，B/A	$t_{s\Phi}(CS)$	読み出しまたは書き込みサイクルの制御信号のセットアップ時間	２８０		ns	
D_0-D_7	$t_{DR}(D)$	$\overline{RD}$の立ち下がりからデータ出力までの遅延		４３０〔２〕	ns	C_L＝５０pF
	$t_{s\Phi}(D)$	書き込みまたはＭ１サイクルのデータのセットアップ時間	５０		ns	
	$t_{DI}(D)$	ＩＮＴＡサイクルの$\overline{IORQ}$の立ち下がりからデータ出力までの遅延		３４０〔２〕	ns	
	t_F　(D)	ＲＤまたは$\overline{IORQ}$の立ち上がりから出力バッファ・フロートまでの遅延		１６０	ns	
ＩＥＩ	$t_S(IEI)$	ＩＮＴＡサイクルの$\overline{IORQ}$の立ち下がりに対するセットアップ時間	１４０		ns	
ＩＥＯ	$t_{DH}(IO)$	ＩＥＩの立ち上がりからの遅延		２１０〔４〕	ns	C_L＝５０pF
	$t_{DL}(IO)$	ＩＥＩの立ち下がりからの遅延（注１）		１９０〔４〕	ns	
	$t_{DM}(IO)$	$\overline{M1}$の立ち下がりからの遅延（Ｍ１サイクルの直前で割り込みが発生したとき）		３００〔４〕	ns	
$\overline{IORQ}$	$t_{s\Phi}(IR)$	読み出しまたは書き込みサイクルのセットアップ時間	２５０		ns	
$\overline{M1}$	$t_{s\Phi}(M1)$	ＩＮＴＡまたはＭ１サイクルのセットアップ時間	２１０		ns	
$\overline{RD}$	$t_{s\Phi}(RD)$	読み出しまたはＭ１サイクルのセットアップ時間	２４０		ns	
$\overline{INT}$	t_D　(IT)	$\overline{STB}$の立ち上がりからの遅延		４９０	ns	
	$t_D(IT3)$	モード３のときのデータ一致からの遅延		４２０	ns	
A_0-A_7，B_0-A_7	t_S　(PD)	モード１のときの$\overline{STB}$の立ち上がりに対するセットアップ時間	２６０		ns	C_L＝５０pF
	$t_{DS}(PD)$	モード２のときの$\overline{STB}$の立ち下がりに対するセットアップ時間		２３０〔４〕	ns	
	t_F　(PD)	モード２のときの$\overline{STB}$の立ち上がりからポート・バス・フロートまでの遅延		２００	ns	
	$t_{DI}(PD)$	モード０のときの書き込みサイクルの$\overline{IORQ}$の立ち上がりからポート出力確定までの遅延		２００〔４〕	ns	
$\overline{A\ STB}$，$\overline{B\ STB}$	t_W　(ST)	$\overline{STB}$のパルス幅（"L"）	１５０〔３〕		ns	
A RDY，	$t_{DH}(RY)$	$\overline{IORQ}$の立ち上がりからの応答時間		t_C+460〔４〕	ns	C_L＝５０pF
B RDY	$t_{DL}(RY)$	$\overline{STB}$の立ち上がりからの応答時間		t_C+400〔４〕	ns	

注　〔１〕 $t_C = t_W(\Phi H) + t_W(\Phi L) + t_r + t_f$

〔２〕負荷容量の５０pF 増加につき、遅延は１０ns 増加します。負荷容量の最大値は２００pFです。

〔３〕モード２のときは、$t_W(ST) > t_S(PD)$となります。

〔４〕負荷容量の１０pF増加につき、遅延は２ns 増加します。負荷容量の最大値は１００pFです。

〔注１〕　デージー・チェーンがＮ段ある場合

$2.5\,t_C > (N-2)\,t_{DL}(IO) + t_{DM}(IO) + t_S(IEI)$ ＋ＴＴＬバッファー遅延を満たさなければなりません。

出力端子測定回路

V_{CC}

測定点

$R_1 = 2.1\ k\Omega$

出力端子

$CR_1 \sim CR_4$；1N914または同等品

CR_1

CR_2

CR_3

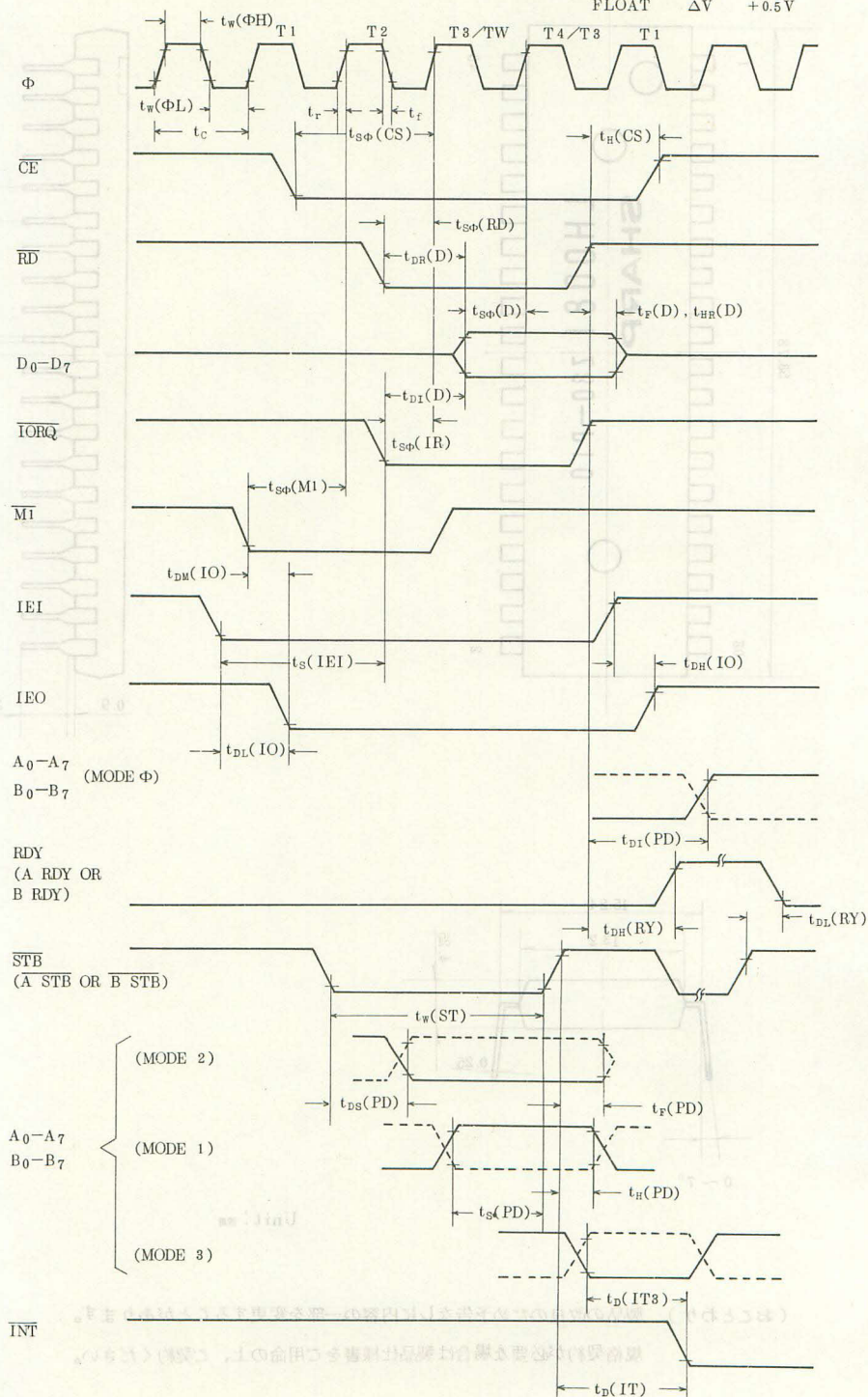
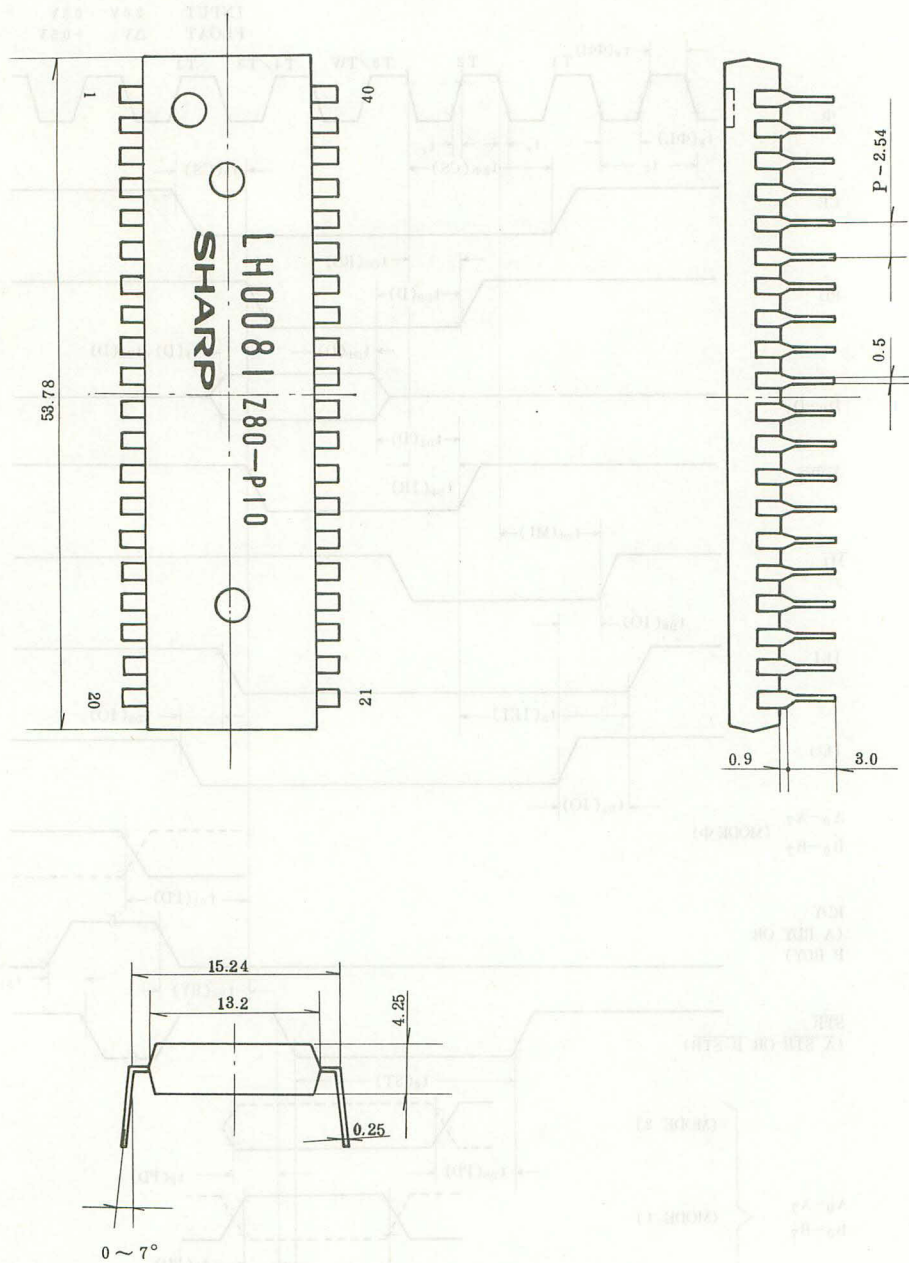
CR_4

C_L

$250\ \mu A$

C_L；すべてのピンに対し50pF

7.4　ＡＣタイミング図

"H" "L"
CLOCK 4.2 V 0.8 V
OUTPUT 2.0 V 0.8 V
INPUT 2.0 V 0.8 V
FLOAT ΔV +0.5 V
Φ
CE
RD
D0−D7
IORQ
M1
IEI
IEO
A0−A7
B0−B7
(MODE Φ)
RDY
(A RDY OR
B RDY)
STB
(A STB OR B STB)
(MODE 2)
(MODE 1)
(MODE 3)
INT
T1
T2
T3/TW
T4/T3
T1

8. 外形寸法図

（おことわり）　製品の改良のため予告なしに内容の一部を変更することがあります。
規格契約が必要な場合は製品仕様書をご用命の上、ご契約ください。

Z-80-CTC
テクニカルマニュアル

3

シャープＺ－８０マイクロコンピュータ[*1]は、マイクロコンピュータ・コンポーネント[*2]とその開発システム、サポート・ソフトウェア[*3]を完備し、容易にシステム設計ができるよう配慮されています。Ｚ－８０マイクロコンピュータ・コンポーネントを採用することにより、別の外部論理回路を付加しなくても高性能のマイクロコンピュータ・システムが得られ、最小限の低コスト標準メモリ[*4]を用いるだけでその目的が達成されます。

ＬＨ－００８２　Ｚ－８０　ＣＴＣ[*5]（以下、Ｚ－８０　ＣＴＣと略します）は、Ｚ－８０システムにおいて、カウンタおよびタイマ機能を与えるプログラム可能な４チャネルを持つカウンタ・タイマ回路です。Ｚ－８０　ＣＴＣのそれぞれ独立した４チャネルは、Ｚ－８０　ＣＰＵ[*6]（ＬＨ－００８０）の指示にしたがって種々のモードや条件で動作します。

1. 特　長

- ○　４つの独立したプログラム可能な８ビット・カウンタ／１６ビット・タイマ回路
- ○　Ｎチャネル・シリコンゲートＥ／Ｄ　ＭＯＳプロセス[*7]
- ○　各チャネルにおいて、カウンタ・モード[*8]とタイマ・モード[*9]の選択可能
- ○　カウンタまたはタイマ状態からのプログラム割り込み[*10]可能
- ○　ダウン・カウンタ[*11]がゼロのとき、時間定数は自動的に再設定され、チャネルは動作を続行
- ○　各チャネルのダウン・カウンタの内容は読み取り可能
- ○　タイマ・モード時に、クロックの１６あるいは２５６分割のプリスケーラ[*12]選択可能
- ○　タイマ起動用トリガおよびカウンタ・モード時のクロック入力の立ち上がり、または立ち下がりの指定可能
- ○　チャネル０～２のＺＣ／ＴＯの出力はダーリントン・トランジスタ[*13]駆動可能
- ○　外部回路を必要としない自動割り込みベクタリング[*14]を行うデージー・チェーン[*15]優先割り込み機能
- ○　＋５Ｖの単一電源、および単相クロック[*16]
- ○　全入出力はＴＴＬコンパチブル[*17]
- ○　パッケージは２８ピンＤＩＰ[*18]

2. 内部構成

Ｚ－８０　ＣＴＣのブロック構成図を図１に示します。Ｚ－８０　ＣＴＣはバス・インターフェース[*19]、内部制御回路、４つのカウンタ・チャネルおよび割り込み制御回路で構成されます。各チャネルは自動割り込みベクタリング用の割り込みベクトルを持っています。割り込み優先順位はチャネル番号の順番であり、チャネル０が最上位です。

---

*1 microcomputer
*2 microcomputer component
*3 support software
*4 memory
*5 Counter Timer Circuit
*6 Central Processing Unit
*7 N-channel silicon gate Enhancement/Depletion type Metal Oxide Semiconductor
*8 counter mode
*9 timer mode
*10 programmable interrupt
*11 down counter
*12 prescaler
*13 darlington transistor
*14 automatic interrupt vectoring
*15 daisy chain
*16 single phase clock
*17 Transistor-Transistor Logic compatible
*18 Dual In line Package
*19 bus interface

図１　Ｚ－８０　ＣＴＣのブロック構成図

図２は各チャネルのブロック構成図を示します。各チャネルは、２つのレジスタ[*7]、２つのカウンタおよび制御回路で構成されています。このレジスタおよびカウンタは、各８ビットの時間定数レジスタ、チャネル制御用レジスタ、プリスケーラおよび読み取り可能なダウン・カウンタです。プリスケーラは、クロックの１６または２５６分割にプログラム可能です。

時間定数レジスタ；８ビットのレジスタで、ＣＰＵによりセットされます。ダウン・カウンタの内容がゼロになるとこのレジスタの内容がダウン・カウンタに再設定されます。

チャネル制御用レジスタ；８ビットのレジスタで、ＣＰＵによりセットされます。チャネルの動作モードおよび条件を選択します。

ダウン・カウンタ；８ビットのカウンタで、プログラム制御によるか、カウンタの内容がゼロになると自動的に、時間定数レジスタの値が設定されます。ＣＰＵはいつでもこのカウンタの内容を読み取ることができます。このカウンタのクロックは、タイマ・モード時にはプリスケーラの出力であり、カウンタ・モード時は外部クロック（ＣＬＫ／ＴＲＧ）です。

プリスケーラ；８ビットのカウンタで、システム・クロック[*8]の１６あるいは２５６分割を行います。この出力はタイマ・モード時に、ダウン・カウンタのクロックとして用いられます。

＊1　data lines　　＊3　CPU bus input／output　　＊5　timeout　　＊7　register

＊2　control lines　　＊4　zero count　　＊6　trigger　　＊8　system clock

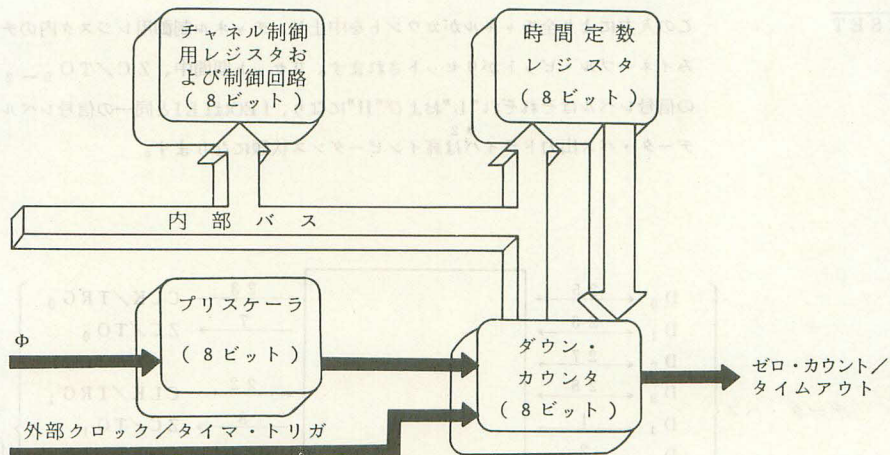



図 2.　チャネルのブロック構成図

3. 端子信号

CLK/TRG_0　　チャネル0用の外部クロック、またはタイマ・トリガ入力（入力端子）

CLK/TRG_1　　チャネル1用の外部クロック、またはタイマ・トリガ入力（入力端子）

CLK/TRG_2　　チャネル2用の外部クロック、またはタイマ・トリガ入力（入力端子）

CLK/TRG_3　　チャネル3用の外部クロック、またはタイマ・トリガ入力（入力端子）

ZC/TO_0　　チャネル0のゼロ・カウント、またはタイム・アウト出力（出力端子）

ZC/TO_1　　チャネル1のゼロ・カウント、またはタイム・アウト出力（出力端子）

ZC/TO_2　　チャネル2のゼロ・カウント、またはタイム・アウト出力（出力端子）

$CS_1 - CS_0$　　チャネル選択入力（入力端子）

2進表示すると、00，01，10および11がそれぞれチャネル0，1，2および3に対応します。

$D_0 - D_7$　　Z－80　CPUのデータ・バス（トライ・ステート[*1]の入出力端子）

$\overline{CE}$　　チップ・イネーブル[*2]入力（入力端子）

Φ　　システム・クロック入力（入力端子）

$\overline{M1}$　　Z－80　CPUからのマシン・サイクル[*3]1信号（入力端子）

$\overline{IORQ}$　　Z－80　CPUからの入出力リクエスト[*4]（入力端子）

$\overline{RD}$　　Z－80　CPUからの読み出しサイクル・ステータス[*5]（入力端子）

IEI　　割り込みイネーブル・イン（入力端子）

IEO　　割り込みイネーブル・アウト（出力端子）

IEIおよびIEOが、割り込み制御における優先順位決定用のデージー・チェーン接続を形成します。

＊1　tri-state　　＊4　input／output request

＊2　chip enable　　＊5　read cycle status

＊3　machine

$\overline{\text{INT}}$	割り込み要求出力(オープン・ドレイン[*1]、出力端子)
$\overline{\text{RESET}}$	この入力により全チャネルがカウントを中止し、チャネル制御用レジスタ内のチャネル割り込みイネーブル・ビットがリセットされます。リセット期間中、ZC/TO_{0-2}および$\overline{\text{INT}}$の信号レベルはそれぞれ"L"および"H"になり、IEOはIEIと同一の信号レベルになります。データ・バス出力ドライバ[*2]は高インピーダンス状態になります。

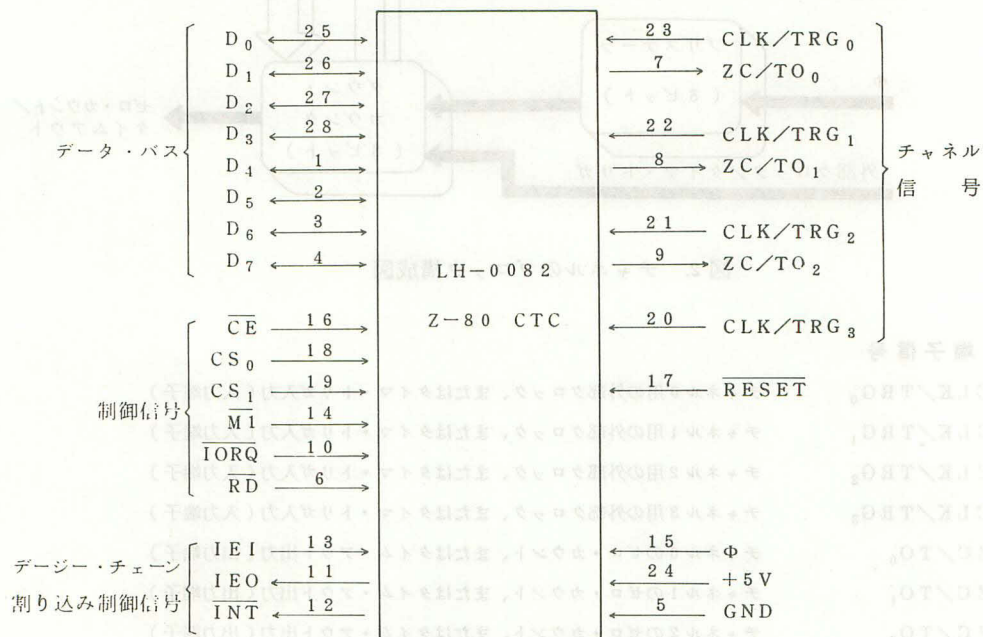



図 3　端子信号

4. 主要タイミング波形

○　書き込みサイクル

チャネル制御情報、時間定数および割り込みベクトルを書き込む際のタイミングを図4に示します。Z−80 CPUにより自動的に挿入される待ち状態(Tw[*])を除き、待ち状態を付加することはできません。Z−80 CTCは明確な書き込み信号入力を持たないため、内部ではRDを書き込み信号に代用します。

＊1 open drain　　＊2 driver

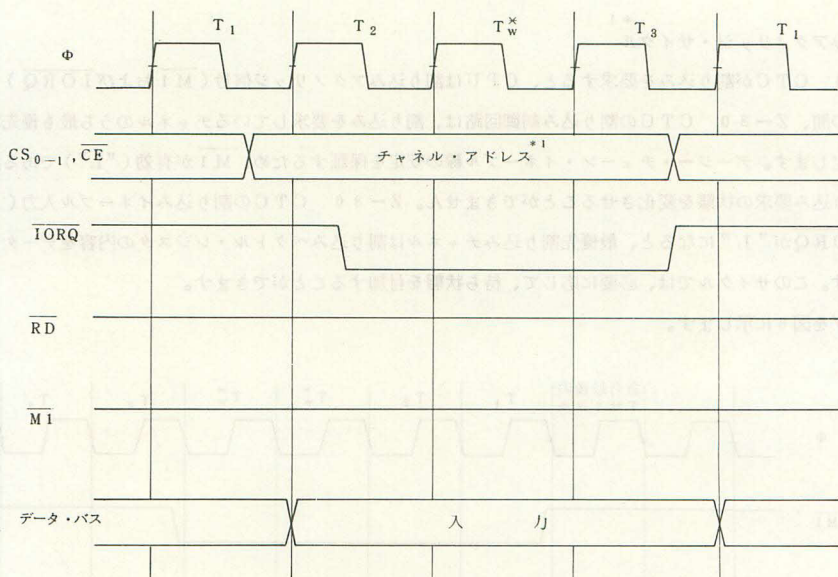



図４　書き込みサイクル

○　読み出しサイクル

チャネル内のダウン・カウンタの内容を読み出すサイクルであり、図５はカウンタ・モードのときのタイミングを示します。データ・バスに読み出されるデータは、この読み出しサイクルのT_2パルス[*2]が立ち上がる直前のダウン・カウンタの内容です。

タイマ・モードの場合にも、データ・バス上のデータはT_2パルスが立ち上がる直前のダウン・カウンタの内容です。Z－８０　CPUにより自動的に挿入される待ち状態（T^*_w）を除き、待ち状態を付加することはできません。

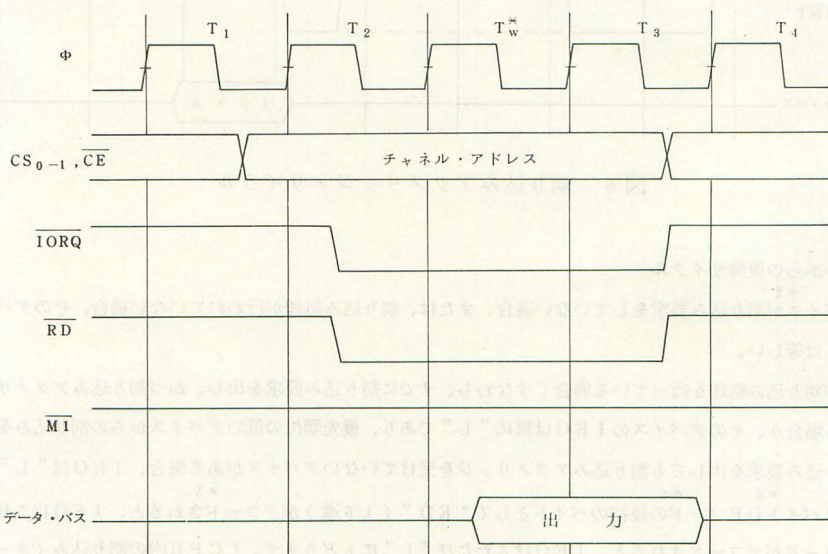



図５　読み出しサイクル

＊1　channel address　　＊2　pulse

○　割り込みアクノリッジ・サイクル[*1]

Ｚ－８０　ＣＴＣが割り込みを要求すると、ＣＰＵは割り込みアクノリッジ信号（$\overline{\text{M1}}$および$\overline{\text{IORQ}}$）を送り出します。この間、Ｚ－８０　ＣＴＣの割り込み制御回路は、割り込みを要求しているチャネルのうち最も優先順位の高いものを決定します。デージー・チェーン・イネーブル線の安定を保証するため、$\overline{\text{M1}}$が有効（"L"）である間、各チャネルは割り込み要求の状態を変化させることができません。Ｚ－８０　ＣＴＣの割り込みイネーブル入力（ＩＥＩ）が"H"で、$\overline{\text{IORQ}}$が"Ｌ"になると、最優先割り込みチャネルは割り込みベクトル・レジスタの内容をデータ・バスに送り出します。このサイクルでは、必要に応じて、待ち状態を付加することができます。

タイミングを図６に示します。

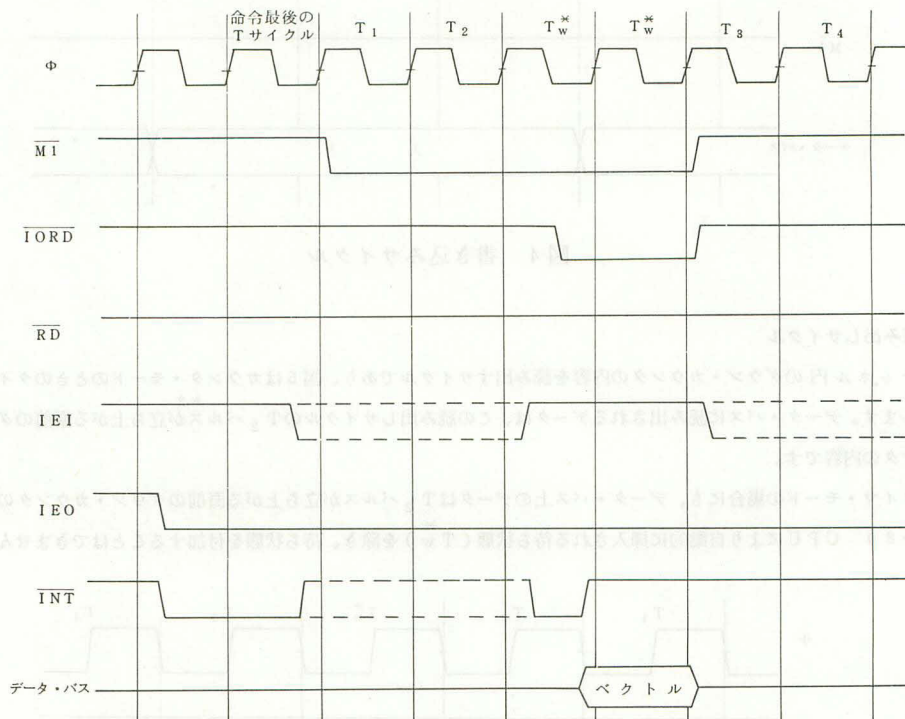



図６　割り込みアクノリッジ・サイクル

○　割り込みからの復帰サイクル

周辺デバイス[*2]が割り込み要求をしていない場合、または、割り込み処理が行われていない場合、そのデバイスのＩＥＯとＩＥＩは等しい。

ＣＰＵが割り込み処理を行っている場合（すなわち、すでに割り込み要求を出し、かつ割り込みアクノリッジを受けとっている場合）、そのデバイスのＩＥＯは常に"Ｌ"であり、優先順位の低いデバイスからの割り込みを禁止します。もし、割り込み要求を出しても割り込みアクノリッジを受けていないデバイスがある場合、ＩＥＯは"Ｌ"になっていますが、2バイト[*3]ＯＰコード[*4]の最初のバイトとして"ＥＤ"（１６進）がデコード[*5]されると、ＩＥＯは"Ｈ"になり、次のＯＰコードがデコードされると、ＩＥＯはふたたび"Ｌ"にもどります。（ＣＰＵ内の割り込みイネーブル・フリ

＊1　interrupt acknowledge cycle
＊2　device
＊3　byte
＊4　operation code
＊5　decode

ップ・フロップ[*1]が0になっているとき、この状態が発生します。）もし、２バイト目のＯＰコードが“４Ｄ”であれば、命令はＲＥＴＩです。

したがって、ＯＰコード“ＥＤ”がデコードされたあと、現在割り込み処理を受けている周辺デバイスだけが、IEI＝“Ｈ”かつＩＥＯ＝“Ｌ”という状態におかれます。このデバイスは、デージー・チェーン中の割り込みアクノリッジを受けとった最優先順位のデバイスです。他のすべての周辺デバイスはＩＥＩ＝ＩＥＯです。次にデコードされたＯＰコードが“４Ｄ”であると、この最優先順位のデバイスは、“割り込み処理中”という状態を解除します。

このサイクルの場合には、待ち状態をＭ１サイクルに挿入することができます。

割り込みからの復帰サイクルのタイミングを図７に示します。

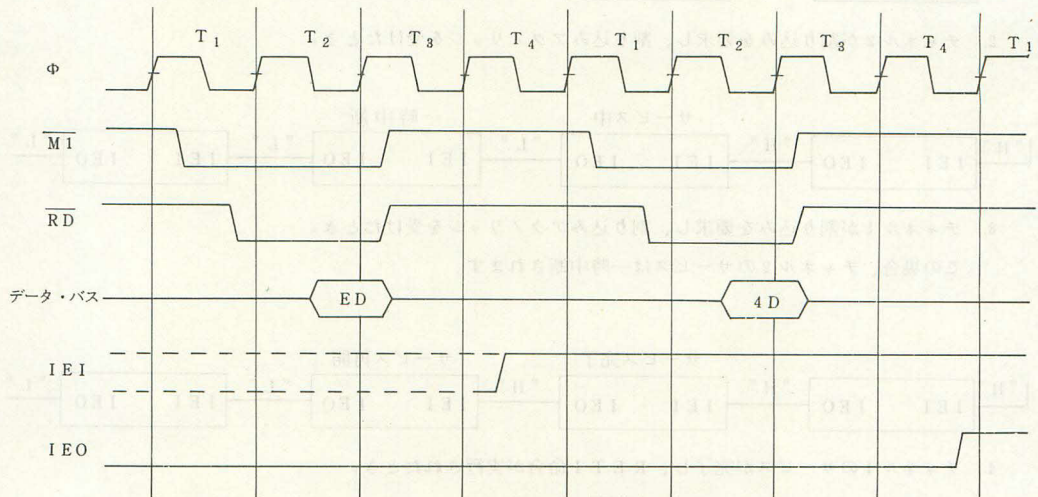



図７　割り込みからの復帰サイクル

○　デージー・チェーン割り込みサービス

図８はＺ－８０　ＣＴＣで起こる典型的なネスト構造の割り込み[*2]順序です。図では、まずチャネル２が割り込みを要求し、サービスを受けることを許されます。チャネル２がサービスを受けている間に、優先順位の高いチャネル１が割り込みを要求すると、チャネル２のサービスは一時中断され、チャネル１がサービスを受けることを許されます。チャネル１のサービス・ルーチン[*3]の実行が完了すると、ＲＥＴＩ命令を実行することにより、チャネル１にサービスが終了したことを知らせます。このとき、チャネル２のサービスが再開されます。

＊1　interrupt enable flip-flop　　＊2　nested inturrupt　　＊3　service routine

優　先　順　位

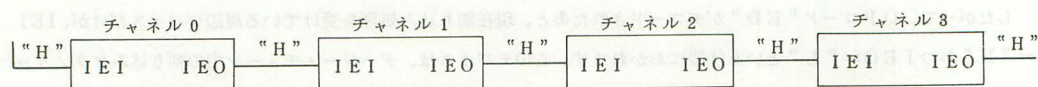
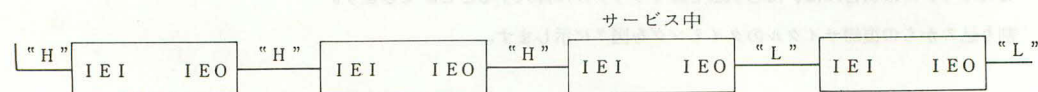
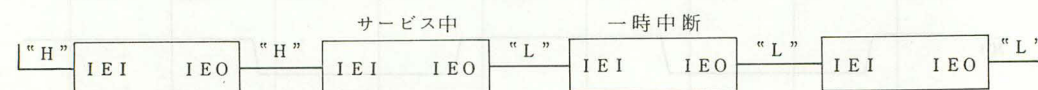
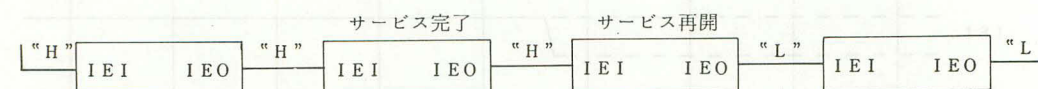
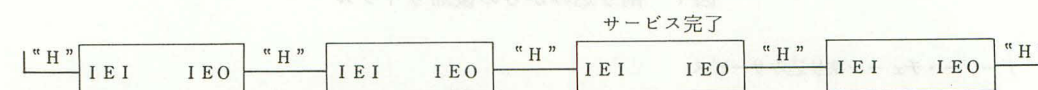
高 ← → 低

1. 割り込み前のデージー・チェーン。

2. チャネル 2 が割り込みを要求し、割り込みアクノリッジを受けたとき。

3. チャネル 1 が割り込みを要求し、割り込みアクノリッジを受けたとき。
この場合、チャネル 2 のサービスは一時中断されます。

4. チャネル 1 のサービスが完了し、RETI 命令が実行されたとき。
この場合、チャネル 2 のサービスが再開されます。

5. チャネル 2 のサービスが完了し、RETI 命令が実行されたとき。

図 8　デージー・チェーン割り込みサービスの説明図

○　カウンタ動作とタイマ動作

カウンタ・モードにおいては、ＣＬＫ入力パルスの立ち上がり、または立ち下がりエッジでダウン・カウンタが動作します。このＣＬＫ入力パルスは非同期入力であり、パルスの最小幅を保証しなければなりません。カウンタはΦの立ち上がりに同期しており、ＣＬＫ入力パルスの入力後、最初のΦの立ち上がりでカウンタを動作させようとする場合、ＣＬＫ入力パルスは必要なセットアップ時間[*1]を満たすものでなければなりません。

タイマ・モードにおいては、ＴＲＧ入力パルスの立ち上がり、または立ち下がりによりプリスケーラの動作を開始させることができます。カウンタ・モードの場合と同様に、このＴＲＧ入力パルスは非同期入力であり、パルスの最小幅を保証しなければなりません。したがって、ＴＲＧ入力パルスの入力後、最初のΦの立ち上がりでプリスケーラを起動しようとする場合、ＴＲＧ入力パルスは必要なセットアップ時間を満たすものでなければなりません。プリスケーラはΦの立ち上がりで動作します。

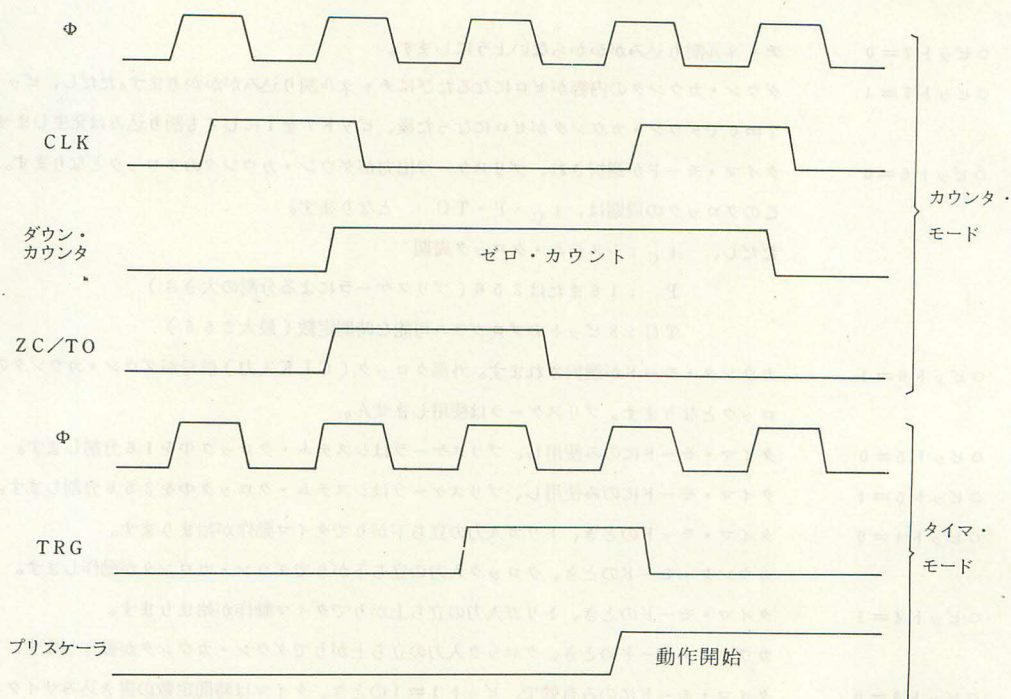



図９　カウンタ動作とタイミング

＊1　setup time

5. 動作条件のプログラム

○　動作モードの選択

チャネルの動作モードを選択するとき、ビット０を１にしたチャネル制御ワードをチャネル制御レジスタに書き込まなければなりません。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
割り込みイネーブル	モード	プリスケーラ選択	エッジの選択	トリガの有無	時間定数の有無	リセット	1

D_3およびD_5はタイマ・モードのときだけ使用します。

○ビット７＝０　　チャネル割り込みがかからないようにします。

○ビット７＝１　　ダウン・カウンタの内容がゼロになるたびにチャネル割り込みがかかります。ただし、ビット７＝０でダウン・カウンタがゼロになった後、ビット７を１にしても割り込みは発生しません。

○ビット６＝０　　タイマ・モードが選択され、プリスケーラ出力がダウン・カウンタのクロックとなります。

このクロックの周期は、t_C・P・TC　となります。

ただし、　t_C；システム・クロック周期

P　；１６または２５６（プリスケーラによる分割の大きさ）

TC；８ビットのプログラム可能な時間定数（最大２５６）

○ビット６＝１　　カウンタ・モードが選択されます。外部クロック（ＣＬＫ入力）信号がダウン・カウンタのクロックとなります。プリスケーラは使用しません。

○ビット５＝０　　タイマ・モードにのみ使用し、プリスケーラはシステム・クロックΦを１６分割します。

○ビット５＝１　　タイマ・モードにのみ使用し、プリスケーラはシステム・クロックΦを２５６分割します。

○ビット４＝０　　タイマ・モードのとき、トリガ入力の立ち下がりでタイマ動作が始まります。

カウンタ・モードのとき、クロック入力の立ち下がりでダウン・カウンタが動作します。

○ビット４＝１　　タイマ・モードのとき、トリガ入力の立ち上がりでタイマ動作が始まります。

カウンタ・モードのとき、クロック入力の立ち上がりでダウン・カウンタが動作します。

○ビット３＝０　　タイマ・モードにのみ有効で、ビット１＝１のとき、タイマは時間定数の書き込みサイクルの次のマシン・サイクルのT_2の立ち上がりから動作を始めます。ビット１＝０のとき、タイマは、この制御情報の書き込みサイクルの次のマシン・サイクルのT_1の立ち上がりから動作を始めます。

○ビット３＝１　　タイマ・モードにのみ有効で、時間定数の書き込みサイクルの次のマシン・サイクルのT_2の立ち上がり後に入力された外部トリガ入力によりタイマが動作を始めます。

トリガ入力がセットアップ時間を満たすときは２つ目のΦの立ち上がりから、また満たさないときは３つ目のΦの立ち上がりからプリスケーラが動作を始めます。時間定数の書き込み以前に外部トリガ入力が加えられると、ビット３＝０の場合と同じになります。

○ビット２＝０　　チャネル制御情報のあとに、時間定数の書き込みがないことを示します。ただし、チャネルがリセット状態にあり、そのあと最初に与える制御情報においては、このビットを０にすることができません。

○ビット2＝1　　チャネル制御情報のあとに、時間定数の書き込みがあることを示します。ダウン・カウンタの動作中に時間定数の書き込みが行われた場合、時間定数レジスタには新しい時間定数がセットされますが、カウントはそのまま続行されます。そして、ゼロ・カウントになったときに初めて新しい時間定数が使用されます。

○ビット1＝0　　チャネルはダウン・カウンタとして動作を行います。

○ビット1＝1　　ダウン・カウンタとしての動作を停止させます。ビット2＝1のとき、時間定数が書き込まれたあと動作を再開します。

ビット2＝0のとき、新しい制御情報を書き込むまで、チャネルは動作しません。

○　時間定数の書き込み

8ビットの時間定数は、ビット2＝1としたチャネル制御情報に続いて、時間定数レジスタに書き込まれます。"00"（16進）は時間定数256を意味します。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
TC_7	TC_6	TC_5	TC_4	TC_3	TC_2	TC_1	TC_0

○　割り込みベクトルの書き込み

割り込みをかけているチャネルは、Z−80　CPUに割り込みベクトルを与えることが必要です。CPUはこのベクトルを使用して、割り込みサービス・ルーチンの番地を作ります。割り込みアクノリッジ・サイクルにおいて、サービスを要求しているチャネルのうち最も優先順位の高いチャネルがCPUにベクトルを与えます。必要な割り込みベクトルは、D_0＝0としてチャネル0の割り込みベクトル・レジスタに書き込まれます。この際使用されるのはD_3〜D_7であり、D_1およびD_2は無視されます。Z−80　CTCが割り込みに応答するとき、ベクトルのうちD_3〜D_7は割り込みベクトル・レジスタの値であり、D_1およびD_2は割り込みを要求したチャネルのうち最も優先順位の高いチャネルの2進符号です。割り込みサービス・ルーチンの開始番地を指示する間接番地が偶数番地であるため、D_0には0がセットされます。チャネル0が最も優先順位の高いチャネルです。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	X	X	0

D_2	D_1	チャネル
0	0	0
0	1	1
1	0	2
1	1	3

6. 絶対最大定格

項目	記号	定格	単位
入力電圧	V_{IN}	−0.3〜＋7	V
出力電圧	V_{OUT}	−0.3〜＋7	V
動作温度	T_{opr}	0〜＋70	℃
保存温度	T_{stg}	−65〜＋150	℃

7. 電気的特性

7.1 DC特性

(T_a＝0℃〜＋70℃, V_{CC}＝＋5V±5%)

記号	項目	最小値	最大値	単位	測定条件
V_{ILC}	クロック“L”入力電圧	−0.3	0.45	V	
V_{IHC}	クロック“H”入力電圧	$V_{CC}-0.6$	$V_{CC}+0.3$	V	
V_{IL}	“L”入力電圧	−0.3	0.8	V	
V_{IH}	“H”入力電圧	2.0	V_{CC}	V	
V_{OL}	“L”出力電圧		0.4	V	$I_{OL}=2mA$
V_{OH}	“H”出力電圧	2.4		V	$I_{OH}=-250\mu A$
I_{CC}	消費電流		120	mA	$t_C=400ns$
I_{LI}	入力リーク電流		10	μA	$V_{IN}=0V \sim V_{CC}$
I_{LOH}	トライステート出力リーク電流		10 *	μA	$V_{OUT}=2.4V \sim V_{CC}$
I_{LOL}	トライステート出力リーク電流		−10 *	μA	$V_{OUT}=0.4V$
I_{OHD}	ダーリントン駆動電流	−1.5 *		mA	$V_{OH}=1.5V$ ZC/TO_0〜ZC/TO_2に適用

＊流入電流を正、流出電流を負とします。

7.2 端子容量

(T_a＝＋25℃, f＝1MHz)

記号	項目	最大値	単位	測定条件
C_Φ	クロック入力容量	25	pF	被測定端子以外の全ての端子は接地
C_{IN}	入力容量	5	pF	
C_{OUT}	出力容量	10	pF	

7.3　AC特性　　　　　　　　　　　　　　　　　　　　　　　　（T_a＝0℃〜＋70℃，V_{cc}＝+5V±5％）

信号	記号	パラメータ	最小値	最大値	単位	備考
Φ	t_C	クロック周期	400	〔1〕	ns	
	t_W(ΦH)	クロック・パルス幅(“H”)	170	2000	ns	
	t_W(ΦL)	クロック・パルス幅(“L”)	170	2000	ns	
	t_r, t_f	クロック立ち上がり・立ち下がり時間		30	ns	
	t_H	ホールド時間	0		ns	
CS, $\overline{CE}$	$t_{S\Phi}$(CS)	読み出し、または書き込みサイクルの制御信号のセットアップ時間	160		ns	
D_0—D_7	t_{DR}(D)	$\overline{RD}$の立ち下がりからデータ出力までの遅延		480	ns	〔2〕
	$t_{S\Phi}$(D)	書き込み、またはM1サイクルのデータのセットアップ時間	60		ns	
	t_{DI}(D)	INTAサイクルの$\overline{IORQ}$の立ち下がりからデータ出力までの遅延		340	ns	〔2〕
	t_F(D)	$\overline{RD}$の立ち上がりから出力バッファ・フロートまでの遅延		230	ns	
IEI	t_S(IEI)	INTAサイクルの$\overline{IORQ}$の立ち下がりに対するセットアップ時間	200		ns	
IEO	t_{DH}(IO)	IEIの立ち上がりからの遅延		220	ns	〔3〕
	t_{DL}(IO)	IEIの立ち下がりからの遅延		190	ns	〔3〕
	t_{DM}(IO)	$\overline{M1}$ の立ち下がりからの遅延（M1サイクルの直前で割り込みが発生したとき）		300	ns	〔3〕
$\overline{IORQ}$	$t_{S\Phi}$(IR)	読み出し、または書き込みサイクルのセットアップ時間	250		ns	
$\overline{M1}$	$t_{S\Phi}$(M1)	INTA、またはM1サイクルのセットアップ時間	210		ns	
$\overline{RD}$	$t_{S\Phi}$(RD)	読み出し、またはM1サイクルのセットアップ時間	240		ns	
$\overline{INT}$	t_{DCK}(IT)	CLK／TRGの立ち上がりからの遅延		2t_C(Φ) +200	ns	カウンタ・モード
	$t_{D\Phi}$(IT)	Φの立ち上がりからの遅延		t_C(Φ) +200	ns	タイマ・モード
CLK／TRG 0—3	t_C(CK)	カウンタ・クロック周期	2t_C(Φ)		ns	カウンタ・モード
	t_r(CK/TR) t_f(CK/TR)	カウンタ・クロックおよびトリガの立ち上がり・立ち下がり時間		50	ns	
	t_S(CK)	即時カウントに要するクロックのセットアップ時間	210		ns	カウンタ・モード
	t_S(TR)	プリスケーラの即時起動に要するトリガのセットアップ時間	210		ns	タイマ・モード
	t_W(CTH)	カウンタ・クロックおよびトリガのパルス幅(“H”)	200		ns	カウンタ・モードおよびタイマ・モード
	t_W(CTL)	カウンタ・クロックおよびトリガのパルス幅(“L”)	200		ns	
ZC／TO 0—2	t_{DH}(ZC)	Φの立ち上がりからZC／TO＝“H”までの遅延		190	ns	カウンタ・モードおよびタイマ・モード
	t_{DL}(ZC)	Φの立ち下がりからZC／TO＝“L”までの遅延		190	ns	

注〔1〕　$t_C = t_W(\Phi_H) + t_W(\Phi_L) + t_r + t_f$

〔2〕　負荷容量の50pF増加につき、遅延は10ns増加します。負荷容量の最大値は、データ・バスが200pFであり、他は100pFです。

〔3〕　負荷容量の10pF増加につき遅延は2ns増加します。負荷容量の最大値は100pFです。

〔4〕　$\overline{RESET}$の入力幅は最低3クロック・サイクル必要です。

出力端子測定回路

Vcc
R1 = 2.1 kΩ
測定点
出力端子
CR1
CR2
CR3
CR4
CL
250 μA
CR1～CR4 ; 1N914 または同等品
CL ; すべてのピンに対し50pF

7.4 ACタイミング図

測定条件は次の通りです。

	"H"	"L"
CLOCK	V_{CC}−0.6V	0.45V
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	△V	±0.5V

8. 外形寸法図

（おことわり） 製品の改良のため予告なしに内容の一部を変更することがあります。

規格契約が必要な場合は製品仕様書をご用命の上、ご契約ください。

SM-B-80D
テクニカルマニュアル

4

目　　　次

1. 概　　要

シャープマイクロコンピュータボードＳＭ－Ｂ－８０Ｄは、１枚のプリント基板上にZ－80CPU、PIO、CTCチップ、OS用ROM、RAM、ユーザ用RAM、シリアルＩ／Ｏインターフェース、パラレルＩ／Ｏインターフェースなどを搭載したものであり、入出力装置を接続するだけでプログラム開発用や制御用のコンピュータとして使用できます。

ＳＭ－Ｂ－８０ＤはＺ－８０マイクロコンピュータ・システムのプログラム開発用の簡易形サポート・ツールとして用意されたものであり、特にそのＯＳ　ＲＯＭ領域にモニタＲＯＭを実装することによりユーザ・プログラムのデバッグを効果的に行うことができます。一方、本ボードは、ＯＳ　ＲＯＭ領域に制御用プログラムを実装することにより直接、機器に組み込み使用できるようになっており、汎用性を考慮した種々の機能を有しています。

Ｚ－８０サポート・ボードとしては、本ボードの他にＲＯＭ／ＲＡＭボードやユニバーサルボードがありますが、これらのボードによりＳＭ－Ｂシステムの拡張が容易に行えるようにボード間でバス信号の共通化を図っています。

モニタ機能によりユーザ・プログラムのロード、パンチ、実行、デバッグなどを行うことができ、また、レジデント・アセンブラやテキスト・エディタもモニタ管理下で使用することができます。これらの場合の入出力装置としてＴＴＹかＲＳ－２３２Ｃ規格の装置を使用できるようになっていますが、ユーザ側で別の入出力装置を定義し、それらをＴＴＹと同じようにモニタで管理することもできます。

2. 特　　長

2．１　ハードウェア

(1)　Ｚ－８０ＣＰＵチップを中心として構成したワンボード・コンピュータです。

(2)　ユーザＲＡＭとして１６ピン・タイプのＲＡＭを使用しており、ソケット実装によりメモリ容量を４Ｋバイトまたは、１６Ｋバイトにできます。

　　４Ｋバイト実装製品：ＬＨ－８Ｈ０１Ａ、１６Ｋバイト実装製品：ＬＨ－８Ｈ０１Ｂ

(3)　ＯＳ　ＲＯＭの容量は４Ｋバイトであり、１Ｋバイト単位に実装できます。

(4)　ＯＳ　ＲＯＭとして、2708タイプ、またはそれとピン互換性のあるＰＲＯＭを実装できます。

(5)　ＯＳ　ＲＯＭとして２Ｋバイト・モニタを用いることによりユーザ・プログラムの実行、デバッグなどができます。（２Ｋバイト・モニタＬＨ－８Ｓ０３Ｐ／Ｅ別売）

(6)　２５６バイトのＯＳ用スクラッチ・パッドＲＡＭ（スタティック）

(7)　ユーザＲＡＭ、ＯＳ　ＲＯＭのベース・アドレスを変更できます。

(8)　Ｚ－８０　ＰＩＯチップによる汎用パラレルＩ／Ｏインターフェース（８ビットのＩ／Ｏポート、２ビットの制御線　各２チャンネル）

(9)　Ｚ－８０　ＣＴＣチップによるカウンタ・タイマ機能　３チャンネル

(10)　ハードウェア割り込み入力２本（ユーザ用ＮＭＩ、ユーザ用ＩＮＴ）

(11)　シリアルＩ／Ｏインターフェースには、ＴＴＹ、ＲＳ－２３２Ｃ規格装置を接続できます。

(12)　ボー・レートを変更できます。110、150、300、600、1200、2400、4800、9600ボー（モニタROM使用時）

(13)　電源投入後、またはリセット後のプログラム開始番地を0000かＥ000のいずれかに選択できます。

(14)　すべてのバス信号線はバッファを内蔵しています。

2．2　ソフトウェア

2Kバイト・モニタの特長（注）

(1) モニタではデータの入出力を次のチャンネルのいずれかを用いて行います。

CI………コンソール入力チャンネル

CO………コンソール出力チャンネル

OI………オブジェクト入力チャンネル

OO………オブジェクト出力チャンネル

SI………ソース入力チャンネル

SO………ソース出力チャンネル

(2) 通常シリアルI／Oインターフェースのドライバ・ルーチンを上記チャンネルに割り当てます。

（例）TTY

(3) 別のI／Oインターフェースのドライバ・ルーチンを用意し、それらを上記チャンネルに割り当てることもできます。　（例）高速パラレルI／O

(4) Z－80　CTCを用いて、8種類のボー・レートを発生できます。

(5) モニタ・コマンド15種

（注）2Kバイト・モニタ　LH－8S03P／E別売

3. 機能概要（モニタROM実装時）

(1) オブジェクト・プログラムのロード

Lコマンド　インテル16進フォーマット

(2) オブジェクト・プログラムのパンチ

Pコマンド　インテル16進フォーマット

(3) ユーザ・プログラムの実行

Gコマンド

(4) ユーザ・プログラムのデバッグ

ソフトウェア・ブレーク・ポイント　B、D、Kコマンド

ユーザ・プログラムのトレース　Tコマンド

ユーザ・プログラムのn命令実行　Sコマンド

メモリ内容の表示と変更　スナップ・ショット

ユーザCPUレジスタの表示　Rコマンド

I／Oポートのテスト　I、Oコマンド

(5) モニタで管理している入出力チャンネル（CI、CO、OI、OO、SI、SO）としてユーザ側で定義したI／Oドライバ・ルーチンを使用できます。

(6) モニタ・プログラム内のI／Oドライバ・ルーチンをユーザ・プログラム内で使用できます。

(7) 汎用パラレルI／Oインターフェース

PIOのA、Bポート（データ線　8ビット×2、制御線　2ビット×2）

ユーザ配線領域（16ピン　DIP　IC　4個実装可能）

50ピン　フラット・ケーブル用コネクタ　1

(8) カウンタ／タイマ

CTC 1個使用、チャンネル1〜3 ユーザ開放

クロック入力 2.4576MHz

(9) リスタート・アドレス切換え機能 0000またはE000

4. コマンド

4.1 記述形式

モニタ・コマンドにおいては、以下に示す文字セット、およびファンクション・キー（印刷出力しない）を使用します。

0〜9、A〜Z

$ ' + , − ・ ／ ; ＝ ↑ 〕

CR LF ETX（CTRL C）

上記以外の文字セット、ファンクション・キーを使用した場合、そのコマンドは無効となり（？を出力する）、再度コマンド待ちとなります。

コマンドは、コマンド識別記号、アーギュメント（argument）、およびターミネータにより構成されており、その一般的な形式は次のいずれかです。

arg_1／

arg_1 , arg_2／　（／以外にLFまたは↑でも可）

arg_1 ; c

arg_1 , arg_2 ; c

ただし、arg_1 , arg_2 はアーギュメント、cはコマンド記号

アーギュメントは、数値、ニーモニック、ロケーション・カウンタ、または、それらを演算子＋、−で結合した式のいずれかです。上記の一般形式において、arg_1 と arg_2 の間はコンマ，で分離しなければなりません。

4．2　コマンド一覧表

機　能	コ マ ン ド	機　能　説　明
メモリ、レジスタの表示	arg_1／nn	arg_1 で指定されるメモリ・アドレス、またはユーザCPUレジスタの内容を２桁の１６進数で／の直後に表示する。表示後ターミネータ待ち。
メモリ、レジスタの変更	arg_1／nn mm	arg_1 で指定されるメモリ・アドレス、またはユーザCPUレジスタの内容を変更する場合に使用する。変更は上記コマンドにより表示された数値に続いて希望する数値（mm）を１６進数で入力し、さらにターミネータを入力することによって行う。
ブレーク・ポイントの設定・解除	arg_1 ；nB	arg_1 がある場合、ブレーク・ポイントの設定を行う。nは０～７で０は省略可能。このコマンドによりアドレス arg_1 に識別番号nのブレーク・ポイントを設定する。arg_1 を省略すると、n番のブレーク・ポイントを解除する。
ブレーク・ポイントの表示	；D	現在設定されているブレーク・ポイントの識別番号とそのアドレス（ブレーク・ポイント・アドレス）をnの順に表示する。
ユーザ・プログラムの実行	arg_1 ；G	arg_1 で指定されるメモリ・アドレスより、プログラム（ユーザ・プログラム）を実行する。arg_1 を省略した場合、現在のPC（ユーザ・CPUレジスタ）で示されるアドレスから実行する。
ポート入力	arg_1 ；nI	arg_1 ＋nで指定されるポートからデータ（１バイト）を読み込み、表示する。nを省略した場合はn＝０と等価である。nは０～２５５の１０進数とする。
ポート出力	arg_1,arg_2；nO	arg_1 ＋nで指定されるポートへ arg_2 で示される１バイトデータを書き込む。nの意見は上記ポート入力の場合と同じ。
ブレーク・ポイントの全解除	；K	現在設定されているすべてのブレーク・ポイントを解除する。
ステップ	arg_1；nS	arg_1 で指定されるメモリ・アドレスより、nステップ実行させ、各ステップごとにPC、AFの内容を印刷出力する。nを省略すると１ステップ動作。
トレース	arg_1；T	arg_1 で指定されるメモリ・アドレスよりトレースする。トレースはCTRL Cのキー入力、またはブレーク・ポイント・アドレスにおいて終了し、コマンド待ちとなる。arg_1 を省略した場合、現在のPCの値よりトレースする。
プログラムのロード	；L	オブジェクト・チャンネルよりインテル標準１６進フォーマットのオブジェクト・プログラムをメモリへロードする。
プログラムのパンチ	arg_1,arg_2；P	arg_1,arg_2 で指定されるメモリ・アドレスの範囲の内容をオブジェクト・チャンネルにインテル標準１６進フォーマットで出力する。
レジスタの表示	；R	すべてのユーザCPUレジスタの内容を表示する。 ；１Rでレジスタ名とその内容を表示する。
メモリ・ブロックの表示	arg_1,arg_2／	arg_1,arg_2 で指定されるメモリ・アドレスの範囲の内容を２桁の１６進数で表示する。
表示モードの指定	；nMまたは arg_1,arg_2；mM	ステップ、トレース、ブレーク・ポイントの各コマンド入力前に表示モードを指定できる。 n（m）　表　示　　　　ただし 0　PC　AF　　　arg_1　メモリ・ブロックの先頭 1　全レジスタ　　arg_2　メモリ・ブロックの最後 2（0）　PC、AFとメモリ・ブロック 3（1）　全レジスタとメモリ・ブロック

5. 基本仕様

5．1　一般基本仕様

項　　目	仕　　　　　　　　様	備　　　　考
CPU	Z－80　CPU　LH－0080	
語　　長	1語　　　　8ビット 命令　　　　8，16，24，32ビット データ　　　8ビット アドレス　　16ビット I／Oアドレス　入力、出力、各8ビット	
最小命令実行時間	1.63μs	4クロック・サイクル （8ビットレジスタ加算 8ビットレジスタ間転送）
CPUクロック	内部クロック（水晶発振）　2.4576MHz　または 外部クロック　0.7～2.5MHz	下限周波数はダイナミックRAMのリフレッシュ・サイクルで決まる。
メ　モ　リ	OS　ROM　LH－2708　最大4個　実装可能 OS　RAM　LH－2111A4　2個　実装 ユーザRAM　LH－4027－3またはLH－4116－3　8個	ソケット実装、ROMは未実装 ソケット実装
メモリ容量	OS　ROM　最大4Kバイト OS　RAM　256バイト ユーザRAM　4Kバイト、または16Kバイト	ユーザRAM 4Kバイト実装製品 LH－8H01A 16Kバイト実装製品 LH－8H01B
メモリ・アドレス	OS　ROM　4Kバイト単位にベース・アドレス設定可能 OS　RAM　FF00～FFFF（固定） ユーザRAM　4Kバイト、または16Kバイト単位にベースアドレス設定可能	ジャンパ端子 ジャンパ端子
パラレルI／O インターフェース	Z－80　PIO　LH－0081　1個使用 8ビット　入出力データ線×2 2ビット　シェーク・ハンド制御線×2 ユーザ配線領域　16ピンDIP IC 4個実装可能 コネクタ　50ピン　フラット・ケーブル用	コネクタJ_1
シリアルI／O インターフェース	UART（8251）　1個使用 TTYインターフェース（20mA電流ループ）および RS－232Cインターフェース コネクタ　26ピン　フラット・ケーブル用	コネクタJ_2
カウンタ／タイマ	Z－80　CTC　LH－0082　1個使用 チャンネル0　　システム使用（ボー・レート作成用） チャンネル1～3　ユーザ開放 クロック入力　　2.4576MHz（406.9ns）	

項　　目	仕　　　　　　　　様	備　　考
I／O ポート・アドレス	ユーザ開放　　00～CF	
	システム使用　　D0～DF	
	システム・リザーブ　E0～FF	
	ただし、D0　　PIO ポートA　データ	
	D1　　PIO ポートA　コントロール	
	D2　　PIO ポートB　データ	
	D3　　PIO ポートB　コントロール	
	D8　　CTC チャンネル　0	
	D9　　CTC チャンネル　1	
	DA　　CTC チャンネル　2	
	DB　　CTC チャンネル　3	
	DC　　UART　データ	
	DD　　UART　コントロール	
	DE　W　システムNMI（N−Delay）	ブレーク・ポイント用
	R　ボー・レート、アドレスEリセット	
	DF　W　システムNMI（Delay）	ステップ・トレース用
	R　リーダ・ステップ	
ボー・レート	8種類切り換え可能 110, 150, 300, 600, 1200, 2400, 4800, 9600	ジャンパ端子
電　　源	＋5V±5％　2.2A max ＋12V±5％　450mA max −12V±5％　150mA max	−5Vは内部電源により発生
動作温度	0℃～50℃	
ボード寸法	270×190×20　　単位 mm	
	J_1　50ピン　フラット・ケーブル・コネクタ （ヒロセ　HIF3−50P−2.54DS相当） J_2　26ピン （同上　HIF3−26P−2.54DS相当） J_3 100ピン　コネクタ　3.175mmピッチ （ケル製　4800−100−135相当）	

関連ソフトウェア（別売）

ソフトウェア名称	形　　　名	備　　考
アセンブラ	LH−8S01P，　LH−8S01E	16KB　RAM必要
エディタ	LH−8S02P，　LH−8S02E	16KB　RAM必要
モニタ	LH−8S03P，　LH−8S03E	
SM−4用クロスアセンブラ	LH−4S04P，　LH−4S04E	16KB　RAM必要
	P：紙テープ版　　E：EPROM版	

5．2　バス信号一覧表

端子番号	信号名（部品面）	端子番号	信号名（配線面）
1	＋5V	51	＋5V
2	＋5V	52	＋5V
3	＋5V	53	＋5V
4		54	
5	＋12V	55	＋12V
6	$*CK/TG_1$	56	
7	$*ZC/TO_1$	57	
8	$*CK/TG_2$	58	
9	$*ZC/TO_2$	59	
10	$*CK/TG_3$	60	
11	*MRESET	61	
12	*DEBUG	62	ø
13	EXCLK	63	XSCLK
14	REC DATA	64	TRANS DATA
15	*DDIS	65	*RENB
16		66	
17		67	
18		68	
19		69	
20	IEI	70	IEO
21		71	
22	$*A_0$	72	$*A_1$
23	$*A_2$	73	$*A_3$
24	$*A_4$	74	$*A_5$
25	$*A_6$	75	$*A_7$
26	$*A_8$	76	$*A_9$
27	$*A_{10}$	77	$*A_{11}$
28	$*A_{12}$	78	$*A_{13}$
29	$*A_{14}$	79	$*A_{15}$
30	*WAIT	80	*BUSRQ
31	*NMIU	81	*INTU
32		82	
33		83	
34	*BUSAK	84	*HALT
35	$*M_1$	85	*RFSH
36	*RD	86	*WR
37	*MREQ	87	*IORQ
38		88	
39		89	
40		90	
41		91	
42	$*D_0$	92	$*D_1$
43	$*D_2$	93	$*D_3$
44	$*D_4$	94	$*D_5$
45	$*D_6$	95	$*D_7$
46	−12V	96	−12V
47		97	
48	GND	98	GND
49	GND	99	GND
50	GND	100	GND

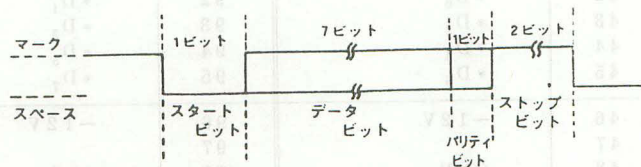
（注）

＊記号は"L"レベルで有効(active)になるという意味であり、信号名の上に――(bar)記号をつけたものに同じ。

空きバスはシステム拡張用としてリザーブ。

５．３　シリアルＩ／Ｏインターフェース信号線説明

信号名	J_2番号	説明
ＧＮＤ	１，１３	接地線
ＲＳ（＋） ＲＳ（－）	６ １６	リーダ・ステップ。リーダ・オン時にＲＳ（＋）、ＲＳ（－）を通じて電流が流れる。（＋１２Ｖで直列抵抗９４Ω）
CARRIER DETECT	１５	20mA SEND信号と同一信号。20mA SEND端子開放時は本端子は＋１２Ｖである。電流ループ使用の場合、UARTの送信データありの状態（スペース）でＨレベル、なしの状態（マーク）でＬレベルとなる。（注１）
20mA SEND 20mA SEND RET	２４ ２５	20mA電源ループでＴＴＹ（ＡＳＲ－３３）を使用する場合、本信号を用いる。UARTの送信データがマークの状態でループ電流オン、スペースの状態でループ電流オフとなる。
20mA REC 20mA REC RET	２２ ２３	20mA電流ループでＴＴＹ（ＡＳＲ－３３）を使用する場合、本信号を用いる。ＴＴＹからの送信データがマークの状態で端子２２はＬレベルに、スペースの状態でＨレベルになっている。
TRANS DATA	３	ＲＳ－２３２Ｃ規格で端末装置からデータが送られてくる場合に用いる入力端子。受信データがマークの状態でＬレベルに、スペースの状態でＨレベルになっている。
REC DATA	５	ＲＳ－２３２Ｃ規格で端末装置へデータを送る場合に用いる出力端子。送信データがマークの状態でＬレベルに、スペースの状態でＨレベルになっている。
ＤＳＲ （Data Set Ready）	１１	UARTの$\overline{\text{DTR}}$信号の反転記号。ＲＳ－２３２Ｃ規格。出力。UART（8251）のコマンドビット１をセットすると$\overline{\text{DTR}}$＝０となる。
ＣＴＳ （Clear To Send）	９	UARTの$\overline{\text{RTS}}$信号の反転信号。ＲＳ－２３２Ｃ規格。出力。UARTのコマンドビット５をセットすると$\overline{\text{RTS}}$＝０となる。
ＤＴＲ （Date Term Ready）	１４	本信号の反転信号がUARTの$\overline{\text{DSR}}$に等しい。ＲＳ－２３２Ｃ規格。入力。UARTの$\overline{\text{DSR}}$をＬレベルにすると、ステイタスビット７がセットされる。
ＲＴＳ （Request To Send）	７	ボード上のチェック端子（記号ＴＴＹ）の結線状態により動作が異なる。 ＴＴＹ　Ａ－Ｃ結線；ＲＴＳ信号　無効 　　　　Ａ－Ｂ結線；ＲＴＳの反転信号＝UARTの$\overline{\text{CTS}}$ 本信号はＲＳ－２３２Ｃ規格で入力信号である。 なお、UARTの$\overline{\text{CTS}}$信号は$\overline{\text{CTS}}$＝Ｌでデータ送信可能。 $\overline{\text{CTS}}$＝Ｈでデータ送信不可（ただしコマンドＴｘＥＮ＝１とする）。

（注１）
UARTの受信または
送信データ
（RxD，TxD）

２ＫＢモニタではデータ送信時偶パリティ、
受信時パリティ無視でデータを処理している。

5．4　ブロック図

5．5　プリント基板寸法図

（おことわり）本資料は製品の改良のため予告なしに内容の一部を変更されることがあります。

SM-B-80T
テクニカルマニュアル

5

目　　　次

シャープマイクロコンピュータボードSMB—80Tは、これからマイクロコンピュータを理解し、実際に使ってみようという方々を対象にした教育用、学習用、ホビー用として、さらにはOEM用としても使用できることを目的に開発されたマイクロコンピュータ・トレーニングボードです。

1. 特　　長

1) プリント配線済みのCPUボード上に、CPU、PIO、メモリ、モニタプログラムを、キーボード上には、8桁7セグメントLED、キースイッチを備えたマイクロコンピュータです。

2) キーボードを使用しないときは、キーボードとのインターフェースに使用しているPIOをユーザが単独で使用できます。さらに、ユーザオプションとしてPIOを1個増設可能で、44ピンコネクタ端子を使用して周辺装置との接続ができます。

3) CPUボード上に100ピンコネクタを設けており、バスドライバを実装することにより外部との接続が容易です。（3.175mmピッチ）

4) ボード内でユーザが使用できるメモリは、ROMが1Kバイト(7055)RAMが3Kバイト(2114)と大容量です。（RAM 1 Kバイト標準装備）

5) カンサスシティ規格のオーディオカセットインターフェースを標準装備しています。（リモート端子による自動、マニアル・スタート/ストップが可能）

6) リスタートアドレスを、モニタプログラム(E000番地)、または、ユーザプログラム(0000番地）にスイッチにて変更できます。このため、RESET・キー操作でユーザプログラムの実行が可能です。

7) Z—80の割り込みのうち、モード0、1、2をユーザに開放しています。

8) CPUボードは、システムに組み込み可能なサイズを採用しています。（270×190mm）SMB—80Dと同寸法です。

2. システム構成

SMB—80Tボードは、CPUボード(270×190mm)と、キーボード(135×190mm)の2枚より構成し、34本のフラットケーブルで接続します。

図2.1に、80Tボードを使用する際の基本的な構成を、図2.2に、システム構成を示します。

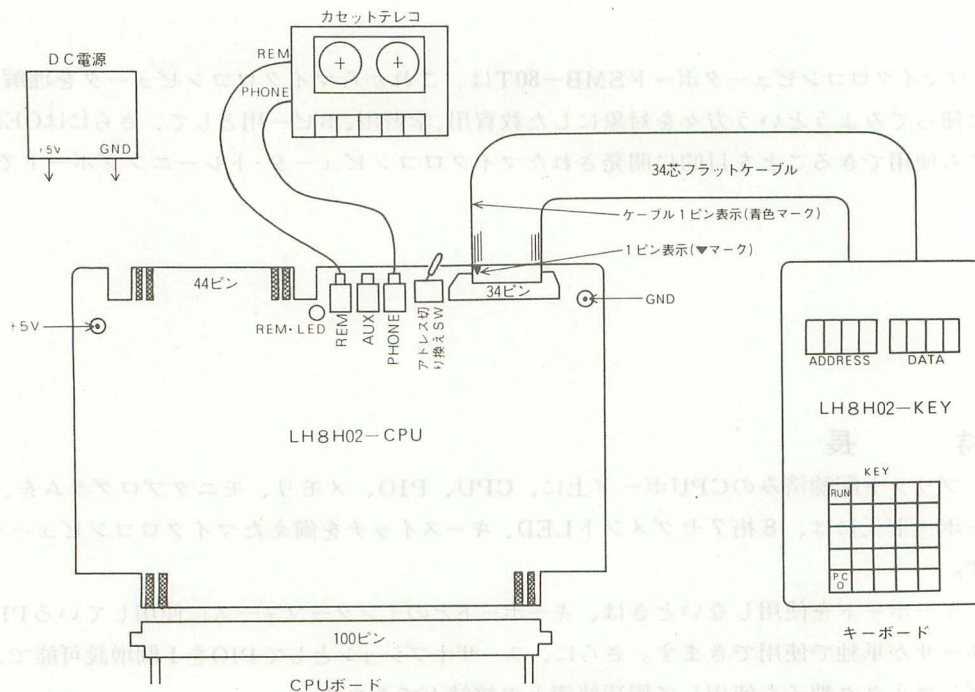



図2.1　SMB−80Tボードの基本的な構成

図2.2　システム構成

3. 仕様概要

C P U	LH－0080　8ビット並列処理プロセッサ
ク ロ ッ ク	2.4576MHz（4.9152MHz 水晶使用）
R O M	MAX　2Kバイト（LH－7055×2個） モニタプログラムを1Kバイトに書き込んで実装。 残り1Kバイトはユーザ用で、ICソケットのみ実装。
R A M	MAX 3.25Kバイト（LH－2114×6個＋LH－2111A 4×2個） モニタプログラム用としてLH 2111A 4×2個(256バイト)をユーザ用としてLH 2114×2個(1Kバイト)を実装。
I/Oポート	LH－0081(PIO)×1個をキーボードとのインターフェースに使用。 ユーザ用としてLH－0081(8ビット×2ポート)×1個オプション(ICソケットのみ実装)。
カセット・インターフェイス	8251×1個をデータの並列↔直列変換に使用。 市販オーディオカセット接続可能。 入／出力端子：REM, AUX, イヤホーン端子 転送速度：300ビット／秒 規格：カンサスシティ規格に準拠
入力装置	データキー、ファンクションキーによる入力。（25キー）
出力装置	7セグメントLEDにより、アドレス、データの16進表示
動作モード	シングルステップ(1命令実行)&オート実行。
モニタプログラム	アドレスE000～E3FF番地の1Kバイトを使用。
コネクタ	100ピンバスライン用コネクタ。 34ピンキーボード接続用コネクタ。 44ピンユーザPIO用コネクタ(コネクタはオプション)
ケーブル	34本フラットケーブル（80cm） REM端子用ケーブル（150cm） AUXまたはイヤホーン用ケーブル（150cm）
電源	＋5V±5％ MAX×2　A(標準構成)
使用温度	0～40℃

4. 機能概要

プログラム機能	0～Fまでの16進データ・キーによる入力。
コンソール機能	メモリの内容表示とその内容の変更。 ユーザレジスタの内容表示と、その内容変更。
デバッグ機能	ユーザプログラムのシングルステップ／オート実行。 ブレークポイント、ブレークカウンタの設定と解除。
オーディオカセット	ユーザが開発したプログラムをオーディオカセットテープへ録音、テープからの再生。(リモート端子により、カセットのスタート/ストップ)

リスタートアドレス	リスタートアドレスを変更可能。 0000：ユーザプログラム開始アドレス E000：モニタプログラム開始アドレス
割り込み	割り込みは、NMIをモニタが使用。 ユーザには、モード0、1、2を開放。
割り込み優先	CPU(NMI)＞PIO1＞PIO2に設定。 ボード外にて、PIO1の上位、PIO2の下位に設定可能。

5. キー配列

RUN	STEP	LOAD INC	STOR DEC	RESET
C	D	E	F	SHIFT
H 8	L 9	A	B	REG REG
BA 4	BC 5	I 6	IF 7	REM ADRS
PC 0	SP 1	IX 2	IY 3	WRITE

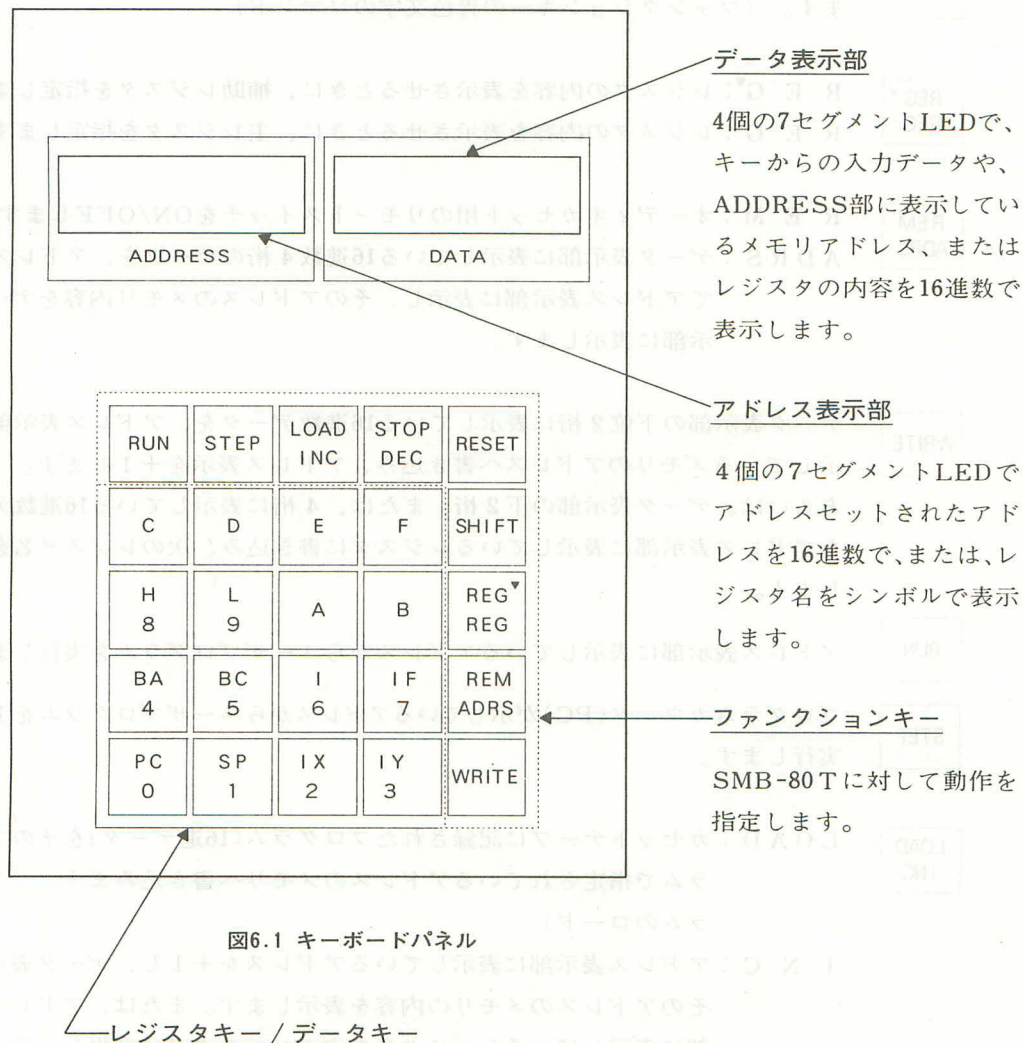
O～F　：データキー
PC～F：レジスタキー
上記以外はファンクションキー

図4.1 キー配列

6. モニタプログラム

SMB－80Tには、プログラムの誤りを捜し出し、修正する機能を有したモニタプログラムを実装しています。以下にモニタプログラムの簡単な説明をします。

6.1 キーボードパネルの構成

図6.1 キーボードパネル

● レジスタキー（上部分とA～Fキー）
　レジスタの指定に使用します。

● データキー（下部分とA～Fキー）
　16進数を入力するときに使用します。

6.2　キーボードスイッチとコマンド

SMB－80Tで使用するキーボードスイッチのコマンドについて説明します。

RESET　プログラム異常(プログラム暴走)によるCPU停止などに対して、システムを初期状態に戻します。

SHIFT　このキー操作後、ダブルファンクション構成キーの上部コマンドが有効になります。（ファンクションキーの青色文字のコマンド）

REG▼ / REG
R　E　G▼：レジスタの内容を表示させるときに、補助レジスタを指定します。
R　E　G：レジスタの内容を表示させるときに、主レジスタを指定します。

REM / ADRS
R　E　M：オーディオカセット用のリモートスイッチをON/OFFします。
ADRS：データ表示部に表示している16進数4桁のデータを、アドレスとしてアドレス表示部に表示し、そのアドレスのメモリ内容をデータ表示部に表示します。

WRITE　データ表示部の下位2桁に表示している16進数データを、アドレス表示部に表示しているメモリのアドレスへ書き込み、アドレス表示を＋1します。
あるいは、データ表示部の下2桁、または、4桁に表示している16進数データをアドレス表示部に表示しているレジスタに書き込み、次のレジスタ名を表示します。

RUN　アドレス表示部に表示しているアドレスからユーザプログラムを実行します。

STEP　プログラムカウンタ(PC)が示しているアドレスからユーザプログラムを1命令実行します。

LOAD / INC
LOAD：カセットテープに記録されたプログラム(16進データ)をそのプログラムで指定されているアドレスのメモリへ書き込みます。（プログラムのロード）
I　N　C：アドレス表示部に表示しているアドレスを＋1し、データ表示部にそのアドレスのメモリの内容を表示します。または、アドレス表示部に表示しているレジスタ名を次のレジスタ名に変更し、データ表示部にそのレジスタの内容を表示します。

STOR / DEC
STOR：アドレス表示部に表示しているアドレスから、データ表示部に表示しているアドレスまでのメモリの内容をカセットテープに記録します。（プログラムのストア）
D　E　C：アドレス表示部に表示しているアドレスを－1し、データ表示部に

そのアドレスのメモリの内容を表示します。または、アドレス表示部に表示しているレジスタ名を前のレジスタ名に戻し、データ表示部にそのレジスタの内容を表示します。

[データキー]　0 ～ F：データ(16進数)の入力に使用します。

[レジスタキー]　アドレス表示部にレジスタ名を表示し、データ表示部にそのレジスタの内容を表示します。

PC	プログラムカウンタ
SP	スタックポインタ
IX	インデックスレジスタX
IY	インデックスレジスタY
BA	ブレークアドレスレジスタ
BC	ブレークカウンタ
I	インターラプトページアドレスレジスタ
IF	インターラプトイネーブルフラグレジスタ
A(A′)	アキュムレータ
F(F′)	フラグレジスタ
B(B′)	Bレジスタ
C(C′)	Cレジスタ
D(D′)	Dレジスタ
E(E′)	Eレジスタ
H(H′)	Hレジスタ
L(L′)	Lレジスタ

(　)内は補助レジスタ

6.3 表　示

16進数とレジスタ名は、7セグメントLEDに次のように表示します。

(1) 16進数キーと表示

キー	表　示	キー	表　示	キー	表　示	キー	表　示
0	o	4	4	8	8	C	C
1	1	5	5	9	9	D	d
2	2	6	6	A	A	E	E
3	3	7	7	B	b	F	F

表6.1 16進数の表示

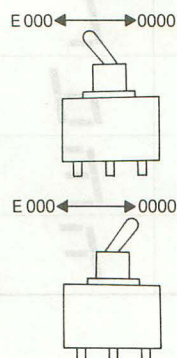
(2) レジスタキーと表示

レジスタを表示させる場合、主レジスタは REG▼/REG キーを、補助レジスタは SHIFT キーを押した後、 REG▼/REG キーを押してから下記レジスタキーで表示させます。

主レジスタ				補助レジスタ	
キー	表示	キー	表示	キー	表示
PC	PC	A	A	A	A'
SP	SP	B	b	B	b'
IX	IH	C	C	C	C'
IY	Iy	D	d	D	d'
I	I	E	E	E	E'
IF	IF	F	F	F	F'
H	h			H	h'
L	L			L	L'

表6.2 レジスタの表示

6.4 アドレス切り換えスイッチ

SMB−80Tは、アドレス切り換えスイッチの設定を変えることによりCPUのリスタートアドレスを次のように変更できます。

リスタートアドレスはE000番地となります。

E000：モニタプログラムの開始アドレス

リスタートアドレスは0000番地となります。

0000：ユーザプログラムの開始アドレス

7. 機　　能

(1) プログラム機能

(2) メモリの内容表示と変更

(3) ユーザレジスタの内容表示と変更

(4) ユーザプログラムのシングルステップ／オート実行

(5) ブレークポイント、カウンタを使用してのプログラム実行

(6) オーディオカセットへのプログラムストア

(7) オーディオカセットよりのプログラムロード

8. 操　作　例

8.1　モニタプログラムスタート

キー操作

アドレス切り換えスイッチ

EOOO側設定

電源スイッチ投入

ADDRESS　　DATA

8.2　データのセット

キー操作

RESET

SP 1　A

ADDRESS　　DATA

データ1A

8.3　アドレスのセット

キー操作

RESET

ADDRESS　　DATA

(注) ××は、O～Fの16進数のうち
どれでも可

アドレス0100番地

0100番地内容

8.4　アドレスのインクリメントとデクリメント

キー操作

キー	ADDRESS	DATA
RESET	0000	0000
LOAD INC	0001 (0000番地＋1)	00XX (0001番地内容)
STOR DEC	0000 (0001番地－1)	XXXX (0001番地内容シフト / 0000番地内容)

8.5　メモリへのデータ書き込み

キー	ADDRESS	DATA
IY 3　E	0100 (書き込みたいアドレス)	XX3E (書き込みデータ)
WRITE	0101 (0100番地＋1)	3EXX (0101番地内容)

8.6　プログラムのオート実行

キー操作

キー	ADDRESS	DATA
PC 0　SP 1　IX 2　IY 3　REM ADRS	0123 (プログラム実行開始アドレス)	23XX (0123番地の内容)
RUN	(実行中は消灯します。)	

8.7　プログラムのシングルステップ実行

キー操作

キー	ADDRESS	DATA
REG▼ REG　PC 0	PC- (プログラムカウンタシンボル)	XXXX (プログラムカウンタ内容)
PC 0　IX 2　PC 0　PC 0　WRITE	SP- (スタックポインタシンボル)	XXXX (スタックポインタ内容)

STEP		ADDRESS: 0201	DATA: XX XX
		1命令実行後のアドレス	アキュムレータ内容 / フラグレジスタ内容

8.8 ブレーク動作

キー操作 ADDRESS DATA

キー操作	ADDRESS	DATA	説明
REG▼ REG, BA 4	bA-	XXXX	ブレークアドレスレジスタシンボル
PC 0, IY 3, PC 0, PC 0, WRITE	bC-	0000	ブレークカウンタシンボル
IX 2, PC 0, WRITE	I -	00XX	インタラプトページアドレスレジスタ
PC 0, SP 1, PC 0, PC 0, REM ADRS	0100	00XX	
RUN	0300	XXXX	0300番地で実行停止 / アキュムレータ内容 / フラグレジスタ内容

(注)引き続きステップ動作を実行するときは、STEPキー操作で実行できます。また、RUNキー操作で0300番地よりプログラムをオート実行できます。

8.9 ユーザーレジスタの内容表示と変更

キー操作 ADDRESS DATA

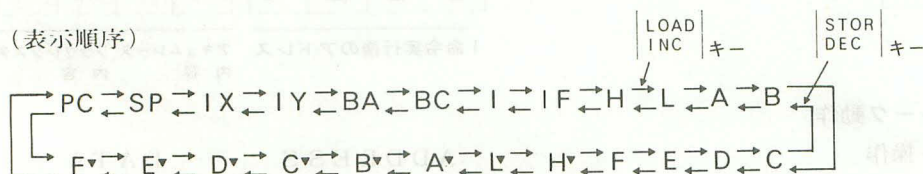
キー操作	ADDRESS	DATA	説明
REG▼ REG, H 8	h -	00XX	Hレジスタシンボル / Hレジスタ内容
SHIFT, REG▼ REG, H 8	h' -	00XX	H▼レジスタシンボル / H▼レジスタ内容
A, B, WRITE	L' -	00XX	L▼レジスタシンボル / L▼レジスタ内容 / H▼レジスタ内容をABに変更

(注)ユーザレジスタは、LOAD INC キー、または、STOR DEC キーで順次表示できます

(表示順序)

LOAD INC キー　STOR DEC キー

PC ⇄ SP ⇄ IX ⇄ IY ⇄ BA ⇄ BC ⇄ I ⇄ IF ⇄ H ⇄ L ⇄ A ⇄ B

F' ⇄ E' ⇄ D' ⇄ C' ⇄ B' ⇄ A' ⇄ L' ⇄ H' ⇄ F ⇄ E ⇄ D ⇄ C

8.10 オーディオカセットへのプログラムのストア(記録)

キー操作　ADDRESS . DATA REM-LED

SHIFT REM ADRS　リモートスイッチON　｝テープを巻き戻して録音状態にしておきます。

SHIFT REM ADRS　リモートスイッチOFF

PC 0 | PC 0 | PC 0 | PC 0 | REM ADRS　　0000 . 00XX

PC 0 | IY 3 | PC 0 | PC 0 　　0000 . 0300

ストア開始アドレス　ストア終了アドレス

SHIFT STOR DEC

ストア実行中は消灯

ストア終了　　0301 . 0300

終了アドレス+1　終了アドレス

8.11 オーディオカセットよりのプログラムのロード(再生)

テープを巻き戻して再生状態にしておきます。

キー操作　ADDRESS　DATA　REM-LED

SHIFT LOAD INC

ロードを実行中は消灯

ロード終了　　0000 . 0300

ロードしたプログラムの開始アドレス　終了アドレス

(おことわり)　本資料は製品の改良のため予告なしに内容の一部を変更することがあります。

SM-B-80D
ユーザーズマニュアル

6

目　　次

1. 概　　要

シャープ　マイクロコンピュータ　ボード　SM−B−80D（LH−8H01 A/B）は、1枚のプリント基板上にZ−80　CPU、PIO、CTCチップ、OS用ROM、RAM、ユーザ用RAM、シリアル I/O インターフェース、パラレル I/O インターフェース等を搭載したものであり、それ自身でまとまったコンピュータ機能を有している。

SM−B−80DはZ−80マイクロコンピュータ・システムのプログラム開発用の簡易形サポート・ツールとして用意されたものであり、特にそのOS　ROM領域にモニタROMを実装することにより、ユーザ・プログラムのデバッグを効果的に行うことができる。一方、本ボードは、OS　ROM領域に制御用プログラムを実装することにより、直接に機器に組み込み使用できるようになっており、汎用性を考慮した種々の機能を有している。

Z−80 サポート・ボードとしては、本ボードの他にメモリ・ボードや汎用 I/O インターフェース・ボードがあるが、これらのボードによりSM−Bシステム の拡張が容易に行えるように、ボード間でバス信号の共通化を図っている。

モニタ機能によりユーザ・プログラムのロード、パンチ、実行、デバッグ等を行うことができ、また　レジデント・アセンブラやテキスト・エディタもモニタ管理下で使用することができる。これらの場合の入出力装置として、TTYかRS−232C規格の装置を使用できるようになっているが、ユーザ側で別の入出力装置を定義し、それらをTTYと同じようにモニタで管理することもできる。

1−1　特　　長

〈ハードウェア〉

(1)　Z−80　CPUチップを中心として構成した、ワン・ボード・コンピュータである。

(2)　ユーザRAMとして16ピン・タイプのRAMを使用しており、ソケット実装によりメモリ容量を4Kバイト、または　16Kバイトにできる。

　　4Kバイト実装製品：LH−8H01A、　16Kバイト実装製品：LH−8H01B

(3)　OS　ROMの容量は4Kバイトであり、1Kバイト単位に実装できる。

(4)　OS　ROMとして、2708タイプ、または　それとピン互換性のあるPROMを実装できる。

(5)　OS　ROMとして2Kバイト・モニタを用いることによりユーザ・プログラムの実行、デバッグ等ができる。（2K　バイト　モニタ　オプション）

(6) ２５６バイトのＯＳ用スクラッチ・パッドＲＡＭ（スタティック）

(7) ユーザＲＡＭ、ＯＳ　ＲＯＭのベース・アドレスを可変できる。

(8) Ｚ－８０　ＰＩＯチップによる汎用パラレルI/Oインターフェース（８ビットのI/Oポート、２ビットの制御線　各２チャンネル）

(9) Ｚ－８０　ＣＴＣチップによるカウンタ・タイマ機能　３チャンネル

(10) ハードウェア割り込み入力　２本（ユーザ用ＮＭＩ、ユーザ用ＩＮＴ）

(11) シリアルI/Oインターフェースには　ＴＴＹ、ＲＳ－２３２Ｃ規格装置を接続できる。

(12) ボー・レートを可変できる。１１０、１５０、３００、６００、１２００、２４００、４８００、９６００ボー

(13) 電源投入後、または　リセット後のプログラム開始番地を００００かＥ０００のいずれかに選択できる。

(14) すべてのバス信号線はバッファを内蔵している。

＜ソフトウェア＞

２Ｋバイト・モニタの特長（２Ｋバイト・モニタ　LH－８Ｓ０３　はオプション）

(1) モニタではデータの入出力を次のチャンネルのいずれかを用いて行う。

ＣＩ…………コンソール入力チャンネル

ＣＯ…………コンソール出力チャンネル

ＯＩ…………オブジェクト入力チャンネル

ＯＯ…………オブジェクト出力チャンネル

ＳＩ…………ソース入力チャンネル

ＳＯ…………ソース出力チャンネル

(2) 通常シリアルI/Oインターフェースのドライバ・ルーチンを上記チャンネルに割り当てる。

例　ＴＴＹ

(3) 別のI/Oインターフェースのドライバ・ルーチンを用意し、それらを上記チャンネルに割り当てることもできる。　　　例　高速パラレルI/O

(4) Ｚ－８０　ＣＴＣを用いて８種類のボー・レートを発生できる。

(5) モニタ・コマンド　１５種

１－２　機能概要

(1) オブジェクト・プログラムのロード

Ｌコマンド、インテル１６進フォーマット

(2)　オブジェクト・プログラムのパンチ

Pコマンド：インテル16進フォーマット

(3)　ユーザ・プログラムの実行

Gコマンド

(4)　ユーザ・プログラムのデバック

ソフトウェア・ブレイク・ポイント：B、D、Kコマンド

ユーザ・プログラムのトレース：Tコマンド

ユーザ・プログラムのn命令実行：Sコマンド

メモリ内容の表示と変更：スナップ・ショット

ユーザCPUレジスタの表示：Rコマンド

I/O ポートのテスト：I、Oコマンド

(5)　モニタで管理している入出力チャンネル（CI、CO、OI、OO、SI、SO）としてユーザ側で定義した I/O ドライバ・ルーチンを使用できる。

(6)　モニタ・プログラム内の I/O ドライバ・ルーチンをユーザ・プログラム内で使用できる。

(7)　汎用パラレル I/O インターフェース

PIOのA、Bポート（データ線　8ビット×2　制御線　2ビット×2）

ユーザ配線領域（16ピン　DIP　IC　4個実装可能）

50ピン　フラット・ケーブル用　コネクタ1

(8)　カウンタ／タイマ

CTC　1個使用、チャンネル1～3　ユーザ開放

クロック入力：2.4576MHz

(9)　リスタート・アドレス切換え機能

1－3　基本仕様

項　　目	仕　　様
CPU	Z－80　CPUチップ
語　　長	1　語　　8ビット 命　令　　8、16、24、32ビット データ　　8ビット アドレス　　16ビット I/O アドレス　入力、出力　　各8ビット

項　　　目	仕　　　　　　　　　　　　　　　様
最小命令実行時間	1.６３μs　８ビット　レジスタ間加算 ８ビット　レジスタ間転送
ＣＰＵ　クロック	2.４５７６　MHz　(水晶発振器　内蔵) 但し、外部クロック　動作可能
メ　　モ　　リ	６４Ｋバイト　　アクセス可能 (１Ｋバイト＝１０２４バイト) ＯＳ　ＲＯＭ　(1)　４Ｋバイト(ソケットにより　１Ｋバイト単位に実装可能) (2)　ベース・アドレス　４Ｋバイト単位に可変 (3)　２７０８タイプのＥＰＲＯＭ、またはそれとピン互換性のあるＰＲＯＭ　実装可能 ＯＳ　ＲＡＭ　(1)　２５６バイト　スタティック　ＲＡＭ (2)　アドレス固定　ＦＦ００(16)〜ＦＦＦＦ(16) ユーザ　ＲＡＭ　(1)　１６Ｋバイトまたは４Ｋバイト　ダイナミックＲＡＭ (2)　１６Ｋバイト、４Ｋバイトどちらでも実装可能 (１６ピン　ソケット　実装) (3)　ベース・アドレス　１６Ｋバイトまたは４Ｋバイト単位に可変
I/O	パラレル I/O (Ｚ−８０　ＰＩＯ　１個使用) ８ビット信号線　　×２ ２ビット制御線　　×２　(シェークハンド可能) シリアル I/O (８２５１　１個使用) モニタで使用(シリアル・クロック　外部供給　可能) カウンタ／タイマ(Ｚ−８０　ＣＴＣ　１個使用) チャンネル１　　　モニタで使用 チャンネル２〜４　ユーザ開放

項　　　目	仕　　　　　　　　様
I/O アドレス	ユーザ開放　　　００(16)　～　ＣＦ(16) システム使用　　Ｄ０(16)　～　ＤＦ(16) 　但し 　　Ｄ０………ＰＩＯ　ポートＡ　データ 　　Ｄ１………　〃　　ポートＡ　コマンド 　　Ｄ２………　〃　　ポートＢ　データ 　　Ｄ３………　〃　　ポートＢ　コマンド 　　Ｄ８………ＣＴＣ　チャンネル　１ 　　Ｄ９………　〃　　　〃　　　２ 　　ＤＡ………　〃　　　〃　　　３ 　　ＤＢ………　〃　　　〃　　　４ 　　ＤＣ………ＵＡＲＴ　データ 　　ＤＤ………　〃　　コマンド 　　ＤＥ………Ｗ　システムＮＭＩ制御（Ｎ－ＤＥＬＡＹ） 　　　　　　　Ｒ　ボー・レート　Ｅリセット 　　ＤＦ………Ｗ　システムＮＭＩ制御（ＤＥＬＡＹ） 　　　　　　　Ｒ　リーダ・ステップ システム・リザーブ　　Ｅ０(16)　～　ＦＦ(16)
ボー・レート	ジャンパ線により８レベル切換可能 １１０、１５０、３００、６００、１２００、２４００、４８００、９６００ Ｂaud
インターフェース	パラレル I/O（コネクタ J_1） 　ＰＩＡ　ポート出力　配線なし 　ユーザ領域　16ピンＩＣ　４個実装可能 シリアル I/O（コネクタ J_2） 　ＴＴＹ（20mA　電流　ループ）及び 　ＲＳ－２３２Ｃ　規格

項　　目	仕　　　　　　　　　　　　　　　　　様
	バス（コネクタ　J_3） すべての入出力信号は負論理とする 入力信号　ＴＴＬ（標準又はＬＳタイプ）インターフェース 出力信号　ＴＴＬ（標準タイプ）インターフェース
電　　源	＋５Ｖ±５％　　2.2Ａmax ＋１２Ｖ±５％　４５０mＡmax －１２Ｖ±５％　１５０mＡmax
動作温度	０℃　～　５０℃
ボード寸法	２７０×１９０×２０　　単位㎜
コネクタ型格	J_1　　５０ピン　フラット・ケーブル・コネクタ J_2　　２６ピン　フラット・ケーブル・コネクタ J_3　１００ピン　コネクタ　3.１７５㎜ピッチ

J_1　ＨＩＦ３－５０Ｐ－2.５４ＤＳ（ヒロセ）相当

J_2　ＨＩＦ３－２６Ｐ－2.５４ＤＳ（ヒロセ）相当

J_3　4800－100－135　（ケル）相当

2. システム構成

2－1　基本システム構成

ワン・ボード・コンピュータ　SM－B－８０Dは１枚のボードと、コンソール用 I/O 装置（例ＴＴＹ）により、手軽にプログラム開発が行えるように用意されたものである。したがって、ボード上には、コンピュータ機能を実現するために必要なリソースは最小限搭載している。すなわち、ＣＰＵチップとその補助回路、ＯＳ用のＲＯＭやＲＡＭ（但し、ＯＳ用ＲＯＭはオプション）、ユーザ用のＲＡＭ、シリアル I/O インターフェースや、パラレル I/O インターフェース等がボド上に実装されており　マイクロコンピュータとしての最小システムを構成している。一方　ボード上のメモリ容量、I/O インターフェースの能力だけでは不足する場合、ボード外部にアドレス、データ、コントロール等の拡張用のバス信号線を取り出せるような構成になっている。図２－１は、SM－B－８０Dと拡張用ボード（開発予定）からなるシステム構成図である。

通常 SM－B－８０Dのコンソール用 I/O 装置として、ＴＴＹかＲＳ－２３２Ｃ規格の装置で用いるが、これらはボード上のシリアル I/O インターフェースによって結線する。しかしながら、シリアル I/O インターフェース経由では低速であり不便な場合、SM－B－８０D上か、またはボード外のパラレル I/O インターフェースを用いて高速 I/O 装置を動作させることも可能である。但し　この場合には、その高速 I/O 装置に適合するインターフェース回路（ハードウェア）とドライバ・ルーチン（ソフトウェア）をユーザ側で用意しなければならない。

SM－B－80 Dの I/O データとしては６種類の形式が定義されており、これらは対応するチャンネル　ＣＩ（Console In）、ＣＯ（Console Out）、ＯＩ（Object In）、ＯＯ（Object Out）、ＳＩ（Source In）、ＳＯ（Source Out）のいずれかを用いて入出力する。ＴＴＹでは、これらのチャンネルは、キー・ボード、タイプ・ヘッド、リーダ等のハードウェアに対応しており、これらのハードウェアを動作させるソフトウェアはシリアル I/O ドライバ・ルーチンとしてモニタ内に内蔵されている。一方高速 I/O 装置によってもこれらのデータ形式を処理することができるが、この場合、前述のようにパラレル I/O インターフェースと、そのドライバ・ルーチンを必要とする。

図２．１のシステム構成では、拡張用のメモリやパラレル I/O インターフェースはSM－B－８０D内のＣＰＵによってアクセスされる。SM－Bシステムでは、ボード内部のメモリや I/O インターフェースをボード外部からもアクセスできるように考慮されており、このような用途のために、データ・バスだけでなくアドレス・バスも双方向性としている。また、メモリや I/O 装置に対するコントロール信号（＊ＭＲＥＱ、＊ＩＯＲＱ、＊ＭＩ、＊ＲＤ、＊ＷＲ、＊ＲＦＳＨ）

(注) 目的のI/O 装置に合った回路を作製する。

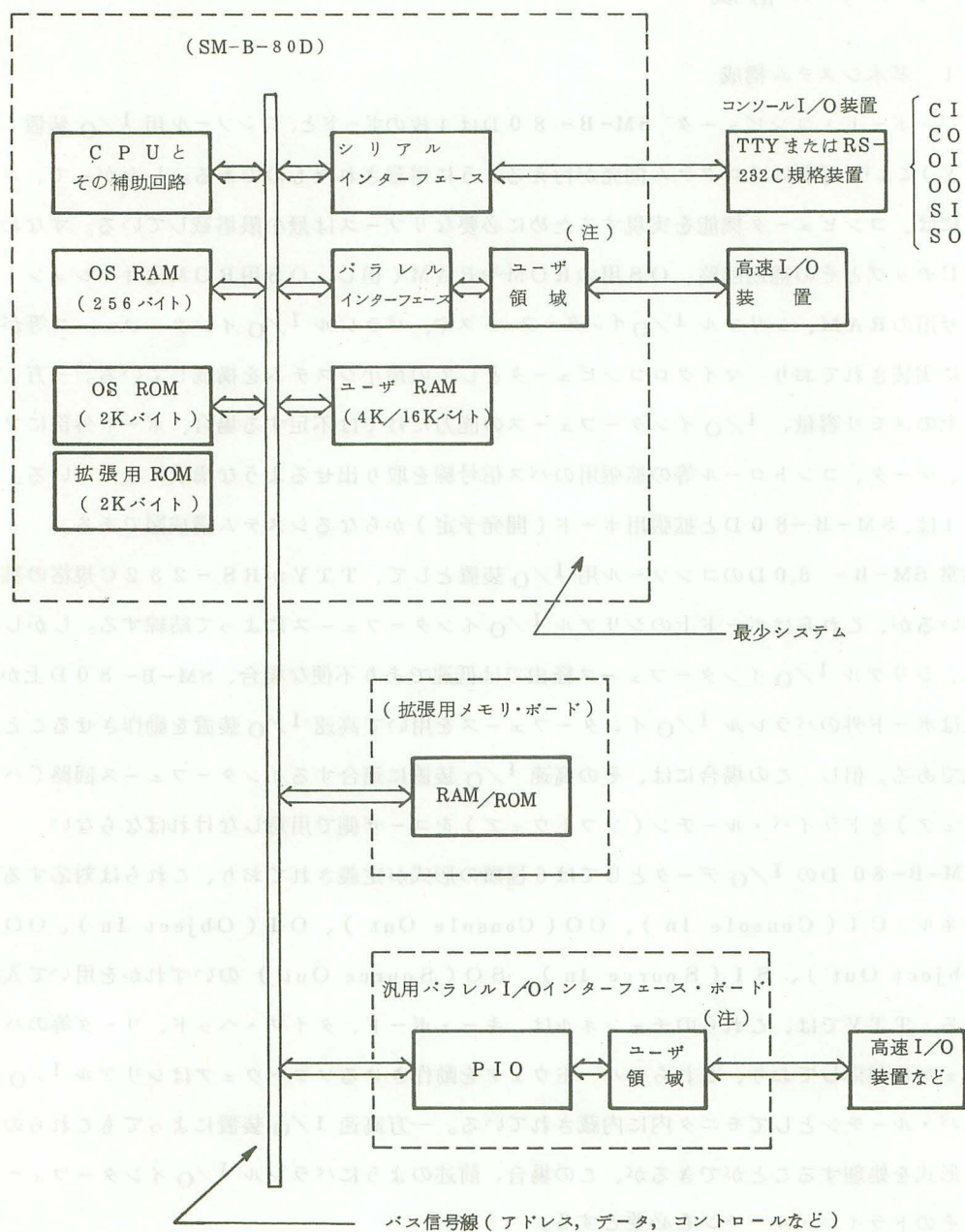



図2．1 シャープ・マイクロコンピュータ・ボード(SM-B)システム構成

も双方向性となっている。この様子を図２．２に示す。このようにすることによって、SM-B-８０D内のメモリや I/O 装置（のバッファ）を外部ＣＰＵによってアクセスしたり、ＤＭＡ転送することが可能になり、ＳＭＢ－８０Ｄ単独で使用する用途だけでなく、より大きいシステムが構成でき、システムの自由度が大きくなる。図２．２は、バス・マスタとなるボードを複数個使用し、その間にバス支配の優先レベルを設ける場合の方法を示したものであり、＊ＢＵＳＡＫをデージー・チェーン接続にすればよい。

２－２　メモリ・マップ

SM-B-８０Ｄのメモリは　①ユーザＲＡＭ　②ＯＳ　ＲＯＭ　③ＯＳ　ＲＡＭにより構成されている。Z-８０　ＣＰＵチップは６４Ｋ（１Ｋ＝１０２４）のメモリ・アドレスを指定できるが、①～③のメモリはこのメモリ・アドレス領域内での指定アドレスが決まっている。以下、これら①～③のメモリのアドレス配分について説明する。

ユーザＲＡＭ

ユーザＲＡＭは１６ピン・タイプのダイナミックＲＡＭを８個使用しており、ボード上のプログラム・ジャンパK_3の切換えにより　４Ｋビット（４０２７相当）、１６Ｋビット（４１１６相当）いずれのタイプのメモリも使用できる。したがって、ボード上のメモリ容量は４Ｋバイトか16Kバイトである。SM-B-８０Ｄでは、ユーザＲＡＭ、ＯＳ　ＲＯＭ共にそのベース・アドレスはボード上のプログラム・ジャンパの配線によって変更できるようになっている。ベース・アドレスは、４Ｋバイトの場合(LH-８Ｈ０１Ａ)、６４Ｋバイトを１６等分することによって求められ、１６Ｋバイトの場合は(LH-８Ｈ０１Ｂ）４等分することによって求められる。この様子を図２．３のメモリ・マップに示す。すなわち４Ｋバイトの場合、そのベース・アドレスは、０００００、１０００、２０００、…………、Ｅ０００、Ｆ０００であり、１６Ｋバイトの場合、００００、４０００、８０００、Ｃ０００のいずれかに設定可能である。

ＯＳ　ＲＯＭ

ＯＳ　ＲＯＭのメモリ容量は４Ｋバイトであり、そのベース・アドレスは４Ｋバイトのユーザ ＲＡＭの場合とまったく同じように設定できる。しかしながら、ＯＳ　ＲＯＭとしてモニタ用ＲＯＭを使用する場合、そのベース・アドレスはＥ０００としなければならない。SM-B-８０Ｄ ではＯＳ　ＲＯＭとして、２７０８タイプのＥＰＲＯＭを使用することを前提としており、４Ｋバイトのメモリ領域はさらに１Ｋバイト単位に分割される。４個のＥＰＲＯＭに配分されるメモリ・アドレスはＯＳ　ＲＯＭのベース・アドレスが決まると１Ｋバイト単位に自動的に決定する.

SM−B−80D

バス・マスタ
となるボード

（注）

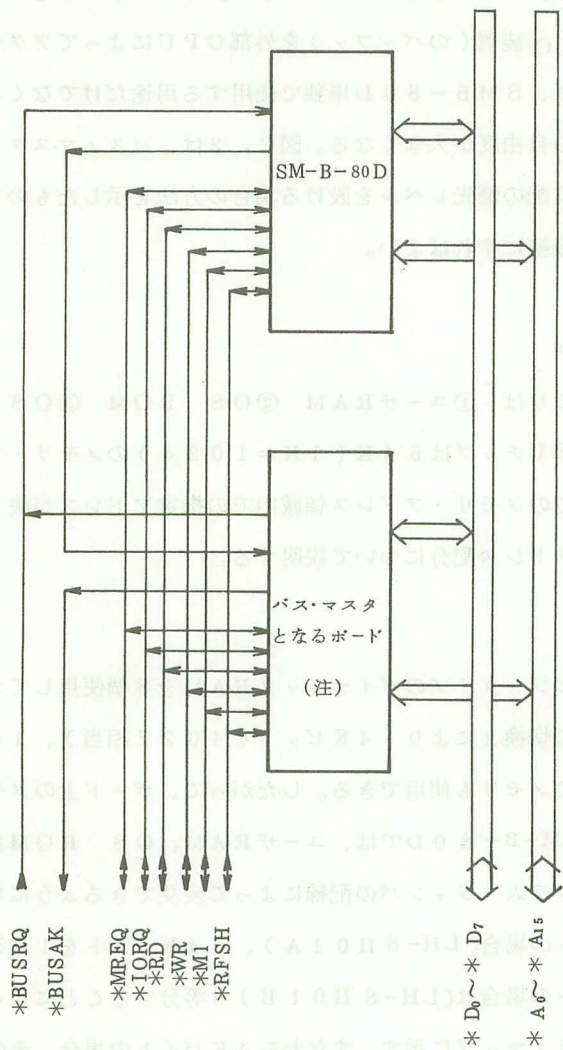
*BUSRQ
*BUSAK
*MREQ
*IORQ
*RD
*WR
*M1
*RFSH
$*D_0 \sim *D_7$
$*A_0 \sim *A_{15}$

（注）　バス・マスタとなるボードとしては、例えばユーザ側で
作製したCPUボードやDMAボードがある。

図2．2　バス共用の結線法

メモリ・アドレス
16Kバイト境界
4Kバイト境界
FFFF
FFFF
OS RAM
F000
E000
D000
C000
C000
B000
A000
9000
8000
8000
7000
6000
5000
4000
4000
3000
2000
1000
0000
0000
FFFF
FF00
OS RAM（固定）
256バイト
FC00
F800
F400
F000
拡張用
ROM #4
EC00
拡張用
ROM #3
E800
E400
OS ROM #2
E000
OS ROM #1
（注）
(1) ユーザRAMのベース・アドレスはボード上のプログラム・ジャンパにより次のように設定できる。
・4Kバイト実装時（LH-8H01A）
0000，1000，2000，………，F000
・16Kバイト実装時（LH-8H01B）
0000，4000，8000，C000
(2) モニタ用ROMのベース・アドレスはボード上のプログラム・ジャンパによりE000とする。
(3) モニタ用ROMを実装しない場合、OS ROMのベース・アドレスはプログラム・ジャンパにより4Kバイト境界で設定できる。
(4) OS RAMは256バイトでアドレスは固定している。

図2．3　メモリ・マップ

ＯＳ　ＲＡＭ

ＯＳ　ＲＡＭのメモリ容量は２５６バイトであり、そのベース・アドレスはハードウェアで固定しており、ＦＦ００である。ＯＳ　ＲＡＭはモニタ・プログラムのスクラッチ・パッド・メモリとして用意されており、モニタ用ＲＯＭ実装時には、通常この領域をユーザ側で使用してはならない。

ＳＭ−Ｂ−８０Ｄをプログラム開発用途に使用する場合、通常、ボード上のユーザＲＡＭのベース・アドレスは００００とし、ＯＳ　ＲＯＭのベース・アドレスはＥ０００とする。ユーザＲＡＭ領域には、レジデント・アセンブラやテキスト・エディタ、あるいは、ユーザ・プログラムをロードする。（レジデント・アセンブラ LH−8S01，テキスト・エディタ LH−8S02はオプション）

なお、①〜③のメモリや、拡張用メモリのアドレス配分において、互いにその領域が重複するような使用をしてはならない。

２−３　リスタートとＮＭＩ制御

Ｚ−８０　ＣＰＵではリスタート・アドレス（リセット終了後のプログラム開始アドレス）は００００であり、ＮＭＩ（Non Maskable Interrupt）の処理開始アドレスは００６６である。ＳＭ−Ｂ−８０ Ｄでは通常モニタ管理下でユーザ・プログラムの実行、デバッグ等を行うために、これらの処理開始アドレスを変更し、各々、Ｅ０００、Ｅ０６６としている。

リスタートが問題となるのは、ＳＭ−Ｂ−８０Ｄへの電源投入後、及びボード上のリセット・スイッチSW_1 を投入した場合であるが、いずれの場合もリスタート・アドレスをＥ０００に強制的にすることにより、プログラムはモニタ・ルーチンから実行を開始する。但し、この動作が行われるのは、スイッチSW_2 をＥ側に倒した場合であり、０側に倒した場合はリスタート・アドレスは００００となる。

ＳＭ−Ｂ−８０Ｄではモニタ・プログラム内でもＮＭＩを使用しており、ユーザ・プログラム内でＮＭＩを使用する場合と区別する必要がある。この区別はＮＭＩの処理開始アドレスをモニタ・プログラムとユーザ・プログラム間で切換えることによって行っている。モニタＮＭＩの処理開始アドレスはＥ０６６であり、ユーザＮＭＩの処理開始アドレスは００６６である。図２．４にリスタート・アドレスとＮＭＩ処理アドレスの制御の考え方を、また、図２．５にそのメモリ・マップを示す。図２．４において、＊ＮＭＩＵは、ユーザ用ＮＭＩのバス信号線であり、ユーザに開放されているものである。＊ＮＭＩＤはモニタで使用しているＮＭＩ信号である。ポート・アドレス　ＤＥ　または、ＤＦに対する出力命令によって＊ＮＭＩＤとＭＮＴＲ信号が出力し

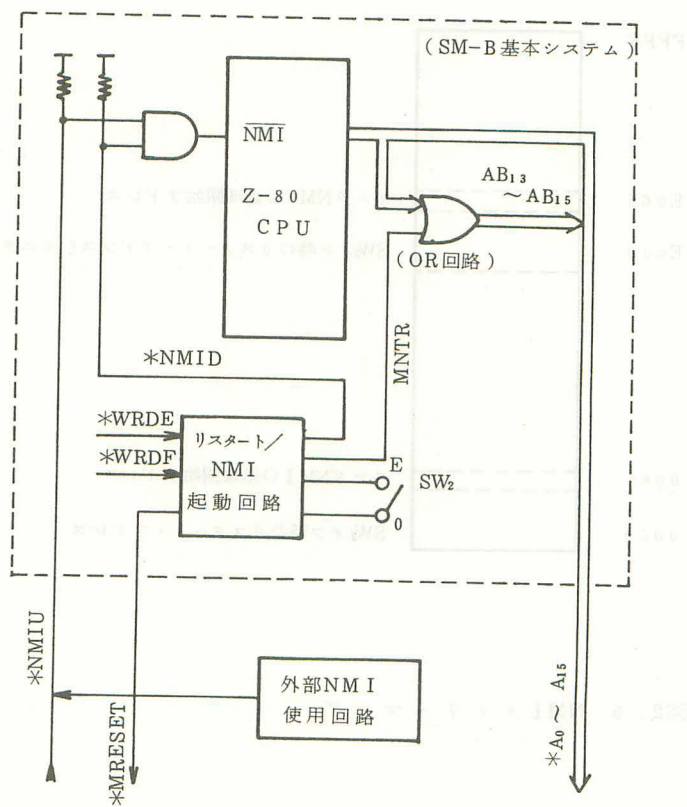



図2．4　リスタート・アドレスとNMI処理アドレスの制御

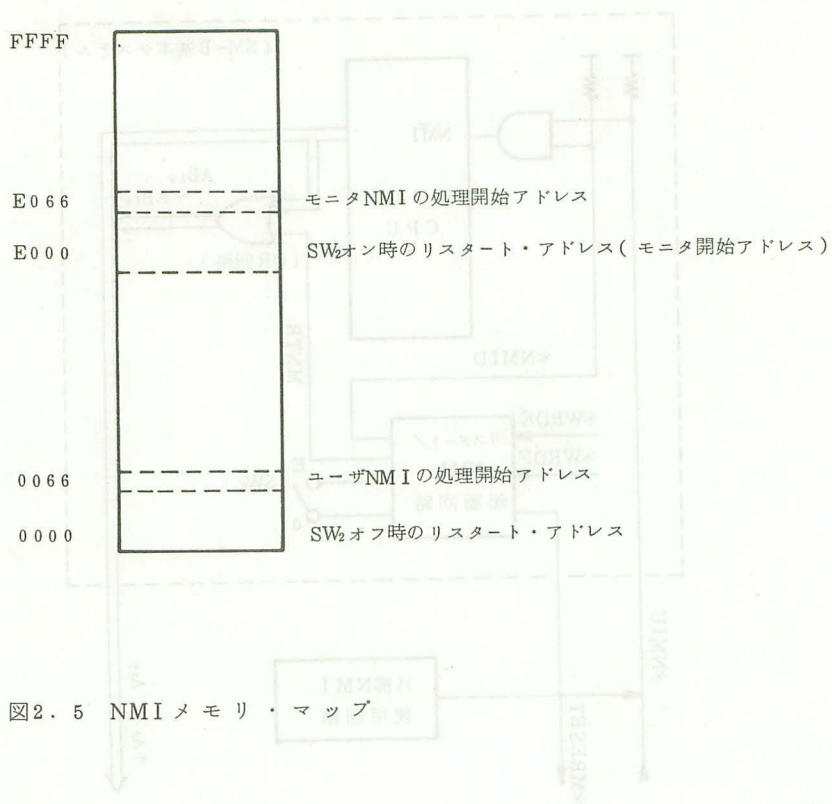
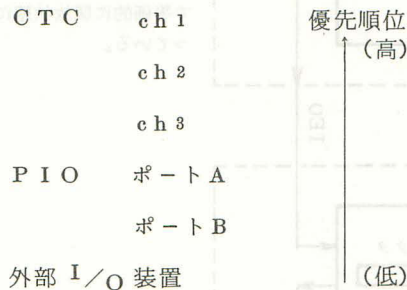



図2．5　NMIメモリ・マップ

プログラム制御はＥ０６６に移る。なおモニタ用ＮＭＩの制御においてはスイッチSW_2の状態は関係しない。一方、電源投入後、または、SW_1オン時には、SW_2がＥ側にある場合に限りＭＮＴＲ信号が出力する。詳細は 3.2.6 参照。

２－４　ＩＮＴ割り込み

図2.6は、ＳＭＢ－８０Ｄ内におけるＩＮＴ関係の信号線を示したものである。モニタ・プログラム内では、ＣＰＵのＩＮＴ信号は使用しておらず、本ボードではＩＮＴ割り込みの使用はすべてユーザ側に委せられている。図において、ＣＴＣのチャンネル０はボー・レート発振器用として、ＳＭＢ－８０Ｄ内で使用されているのでユーザ側での使用はできない。このため、チャンネル０からの割り込みはモニタ・プログラムによりマスクされている。図からもわかるように、ＳＭＢ－８０ＤではＩＮＴ割り込みはモード２での使用を前提としており、ＣＴＣのチャンネル１、２、３、ＰＩＯのポートＡ、Ｂの順にデージー・チェーン接続になっている。また、ＰＩＯのＩＥＯ出力はボード外部へ取り出せるようになっており、図のように外部 I／O インターフェースもデージー・チェーン接続にできる。この場合の割り込み優先順位は次のようになる。

ＩＮＴ割り込みをモード２で使用する場合、次の点に留意しなければならない。まず、電源投入後やリセット後ではＣＰＵのＩＮＴ割り込みはモード０になっているので、モード２にプログラムしなければならない。さらに、各ＩＮＴ割り込みの開始アドレス（下位バイト）を与えるポインタを指定するために、ＣＰＵのＩレジスタとＣＴＣ、ＰＩＯ等の各コントロール・レジスタにベクタをロードしなければならない。

SM－B－８０ Ｄでは、上記のモード２以外にモード０やモード１の割り込みを使用することも可能であるが、これはＣＰＵにＩＮＴ割り込みのモードを指示することによって行う。シリアル・インターフェースの８２５１から割り込みをかけることもできるが、これは通常ジャンパ線K_1により動作しないようにしている。

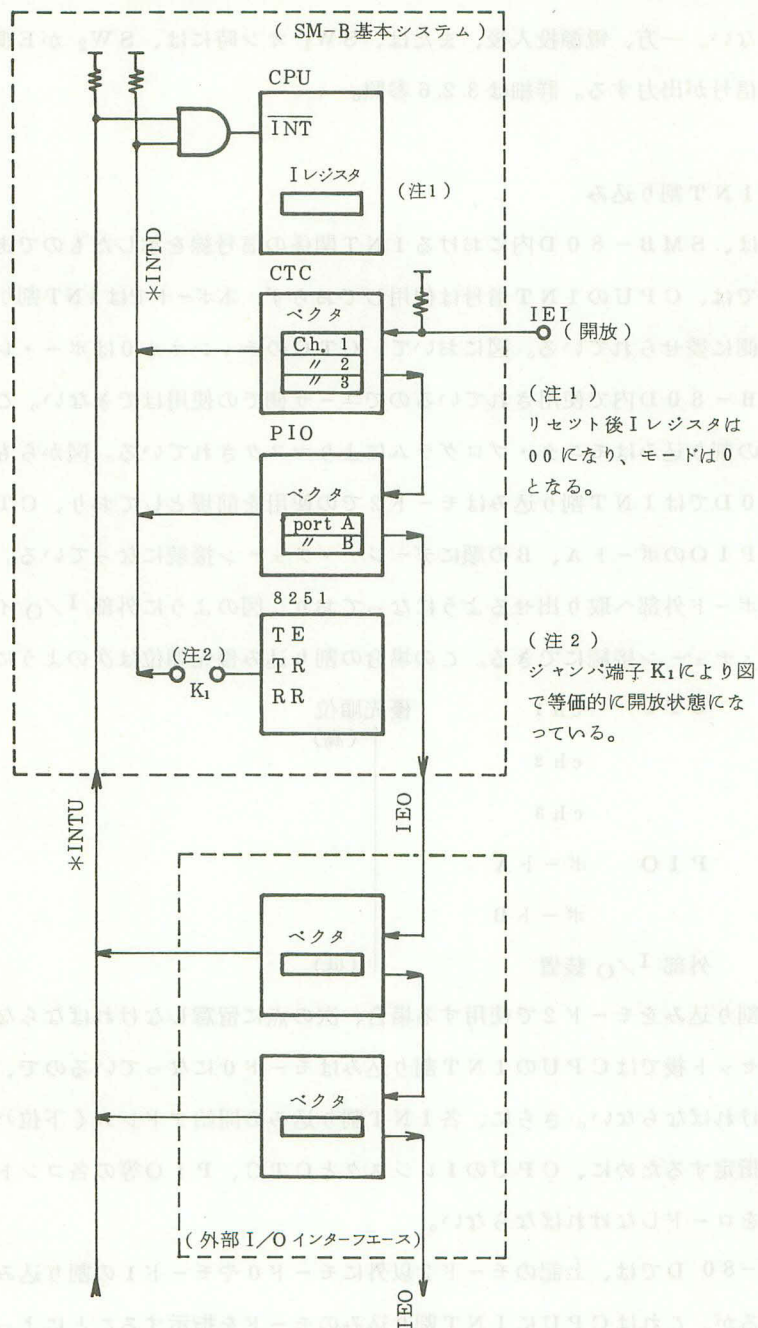



図2．6 $\overline{\text{INT}}$ 割り込みとデージー・チエーン

3. ハードウェア

3－1 構　　成

図3．1は、SM－B－80 Dのブロック図であり、［　　　］で囲まれた部分は各機能単位のブロックを示している。太線はバス信号を示し、細線は1本の信号を示している。各ブロックの動作は次の3．2において述べるので、ここでは全体的な動作について説明する。

SM－B－80DはZ－80　CPUチップを中心として、図に示すように13のブロックにより構成されている。

アドレス・バッファはCPUから出力するアドレス信号をまずバッファし、ボード内の各ブロック、及びボード外部へ供給する。アドレス・バッファを双方向性とすることにより、ボード外部から内部のメモリや I／O バッファをアクセスできるようになっている（これはDMA転送やマルチCPUのときに用いる）。

データ・バッファはデータの双方向性バッファ、及び　レシーバとドライバの制御回路により構成される。レシーバは通常ディスエーブルになっており、レシーバの制御は外部制御信号＊RENBによってのみ可能となる。ドライバはSM－B－80D内のメモリか I／O インターフェースが読み出された場合にイネーブルになり、外部に対してデータを出力する。なお、ドライバは外部制御信号＊DDISによっても制御可能である。

コントロール・バッファは2つに分かれる。一方はCPUへの制御入力に対するバッファであり、＊WAIT、＊BUSRQ、＊NMIU、＊INTU信号をバッファする。他方はCPUからの制御出力に対するバッファである。制御出力のうち＊HALT、＊BUSAKは出力専用でありボード自身の状態を表わす。＊MREQ、＊IORQ、＊RD、＊WR、＊MI、＊RFSHに対するバッファは双方向性であり、ボード外からこれらの制御線を使用できるようになっている。

クロック発振器は水晶発振回路とバッファから成り、CPU、ボード内の各部、及び　ボード外部へクロックを供給している。クロックは外部から供給することもできる。

リセット回路はCPU、及び　各部へのリセット信号を作成する回路であり、パワー・オン・リスタート回路とマニュアル・リセット回路から成る。マスタ・リセット信号＊MRESETは双方向性であり、外部からもSM－B－80 Dをリセットできる。

リスタート／NMI起動回路は2つの機能を有している。一方はパワ・オン・リスタートやマニュアル・リセットやモニタでのNMIの使用時に、アドレスの上位3ビット（$AB_{15\sim13}$）を強制的に1にする信号MNTRを作成する回路であり、他方はモニタ・ルーチンで使用するNM

*MRESET
EXCLK
φ
XT
φ. CLK
*WAIT
*BUSRQ
*NMIU
*INTU
*DEBUG
IEI
XSCLK
SW1
リセット回路
クロック発振器
コントロールバッファ(1)
Z-80 CPU
RESET
WAIT
BUSRQ
INT
NMI
HALT
BUSAK
MREQ
IORQ
RD
WR
M1
RFSH
*MRST
*BUSAK
MNTR
アドレスバッファ
データバッファ
*A0 〜 *A15
*RENB
*DDIS
*D0 〜 *D7
LED
コントロールバッファ(2)
*HALT
*BUSAK
*MREQ
*IORQ
*RD
*WD
*M1
*RFSH
AB0〜AB15
DB0〜DB7
ユーザRAM
OS RAM
OS ROM
アドレスデコーダ
RAM
*OS RAM
SEL
ROM
*UART
*PIO
*CTC
*RDDF
*RDDE
*WRDF
*WRDE
リスタート/NMI起動回路
SW2
カウンタ/タイマ
パラレルI/Oインターフェース
シリアルI/Oインターフェース
S.CLK
J1(50ピン)
J2(26ピン)
PIO用ユーザ領域
TTY 信号線
RS-232C装置信号線
IEO
*CK/TG1〜3
*ZC/TO1〜2
*NMID
*INTD

図3.1 ブロック図

I信号（＊NMID）を作成する回路である。モニターのトレース、ステップ、ブレイク・ポイントの各コマンドではNMI信号を使用している。

アドレス・デコータはメモリ・アドレス・デコーダと$^{I}/_{O}$ポート・アドレス・デコーダに分類できる。メモリ・アドレス・デコーダは　OS、ROM、OS、RAM、ユーザRAMの各選択信号を作成し、$^{I}/_{O}$ポート・アドレス・デコーダはカウンタ／タイマ、パラレル$^{I}/_{O}$インターフェース、シリアル$^{I}/_{O}$インターフェース、ボードのステータスやコマンド・ビットの各選択信号を作成する。

OS　ROMのメモリ容量は４Kバイトであり、１Kバイト単位にソケット実装ができる。通常ROMとしては２７０８タイプのEPROMを使用するが、２７０８とピン互換性のあるPROMも実装できるように電源ラインを考慮してある。モニタを用いる通常の使用法では、モニタ用EPROMをソケット＃１、＃２に実装する。（モニタ　LH－8S03はオプション）

OS　RAMのアドレスは固定しており、FF００～FFFFの範囲の２５６バイトである。OS　RAMとしては、２１１１タイプのスタティックRAM（１８ピン２５６×４ビット）を２個使用する。

ユーザRAMは１６ピン・タイプのダイナミックRAMとRAS／CAS切換え回路、及びデータ・バッファから成る。ダイナミックRAMは４Kビット（４０２７相当）、１６Kビット（４１１６相当）いずれのタイプでも使用できるようにソケット実装になっている。

カウンタ／タイマ回路は、CTCチップとボー・レート設定回路から成る。ボー・レート設定回路は３ビットのプログラム・ジャンパにより８種類のボー・レートを設定できる。CTCはチャンネル０をモニタで使用している。チャンネル０をタイマ・モードで使用することにより、UART用のボー・レート・クロックを作成しているが、ボー・レートの選定は既述の３ビットのジャンパの状態をプログラムにより読み込むことにより行う。

パラレル$^{I}/_{O}$インターフェースはPIOチップとユーザ領域、及び　インターフェース用５０ピン・コネクタから成る。PIOのA、Bポート入出力は何も配線されていない。PIOをどのように使用するかはユーザ側に委されている。ユーザ領域は、１６ピンのDIP　ICを４個実装できる。

シリアル$^{I}/_{O}$インターフェースは　UART（８２５１相当）と、TTY　及び　RS－２３２Cインターフェース回路により構成されている。　UARTに対するシリアル・クロックは既述のようにCTCのチャンネル０から供給しているが、外部からシリアル・クロックを供給することもできる。

3－2　動　　作

3－2－1　アドレス・バッファ

図３．２にアドレス・バッファの回路図を示す。U_{57} ～U_{60} は双方向性バッファ８Ｔ２６である。アドレス信号のうち、ビット０～１１とビット１２～１５の動作は異なる。

アドレス信号A_0 ～A_{11}は＊ＢＵＳＡＫ＝Ｈのときバス信号＊A_0 ～A_{11} として外部へ出力する。また、U_{57} ～U_{59} のレシーバは常にイネーブルになっており、ドライバの出力信号、または　外部からの入力信号はレシーバを経てボード内の各部に供給される。＊ＢＵＳＡＫ＝Ｌのときドライバ出力はトライ・ステートになるが、外部からはアドレス信号を供給できる。

アドレス信号A_{12} ～A_{15} は図に示すように、ＣＰＵの出力とレシーバの出力をワイアドＯＲにしてボード内各部に供給している。U_{60}において、ドライバとレシーバは＊ＢＵＳＡＫにより制御されており、＊ＢＵＳＡＫ＝Ｈのときレシーバ出力がトライ・ステートになり、＊ＢＵＳＡＫ＝Ｌのときドライバ出力がトライ・ステートになる。ボード内の各部に供給されるアドレス信号AB_0 ～AB_{15} は正論理となっており、リスタート／ＮＭＩ起動回路の出力信号ＭＮＴＲがＨレベルのときAB_{13}～AB_{15} は強制的に１になる。

3－2－2　データ・バッファ

図３．３にデータ・バッファの回路図を示す。U_{62}、U_{63} は双方向性バッファ８Ｔ２６である。

U_{62}、$_{63}$ のレシーバはバス信号線＊ＲＥＮＢによって制御され、通常レシーバはデイスエーブルされている。＊ＲＥＮＢ＝Ｌのときレシーバはイネーブルになりボード外部からデータを入力できる。

U_{62}、$_{63}$ のドライバはバス信号線＊ＤＤＩＳ、または　内部ドライバ制御信号により制御される。ドライバがイネーブルになる条件は、＊ＤＤＩＳ＝Ｈ、かつ　内部ドライバ制御信号＝Ｈのときであり、さらに内部ドライバ制御信号がＨレベルとなるのは次のいずれかの場合である。

① ＊ＭＲＥＱか＊ＩＯＲＱがＬレベルであり、かつ、＊M_1、＊ＢＵＳＡＫ、＊ＲＤがいずれもＨレベルのとき。

② ＲＦＳＨ＝Ｈ、かつ　＊ＲＤ＝Ｌ、かつ　＊ＭＲＥＱ＝Ｌであり、SM-B-８０Ｄ上のいずれかのメモリがアクセスされたとき。

3－2－3　コントロール・バッファ

図３．４にコントロール・バッファの回路図を示す。U_{61}、U_{64}は単方向バッファ８Ｔ９７

図3．2　アドレス・バッファ

CPUへ
D0
D1
D2
D3
D4
D5
D6
D7
U62
DI
RO
DB
DE RE
U63
42 *D0
92 *D1
43 *D2
93 *D3
44 *D4
94 *D5
45 *D6
95 *D7
DB0
DB1
DB7
R30
U56
R37
*RENB 65
*DDIS 15
RAM SEL
*ROM SEL
*OS RAM SEL
*MREQ
*RFSH
*RD
*IORQ
*BUSAK
*M1
U38
U42
U37
U65
O/C
U39
U43

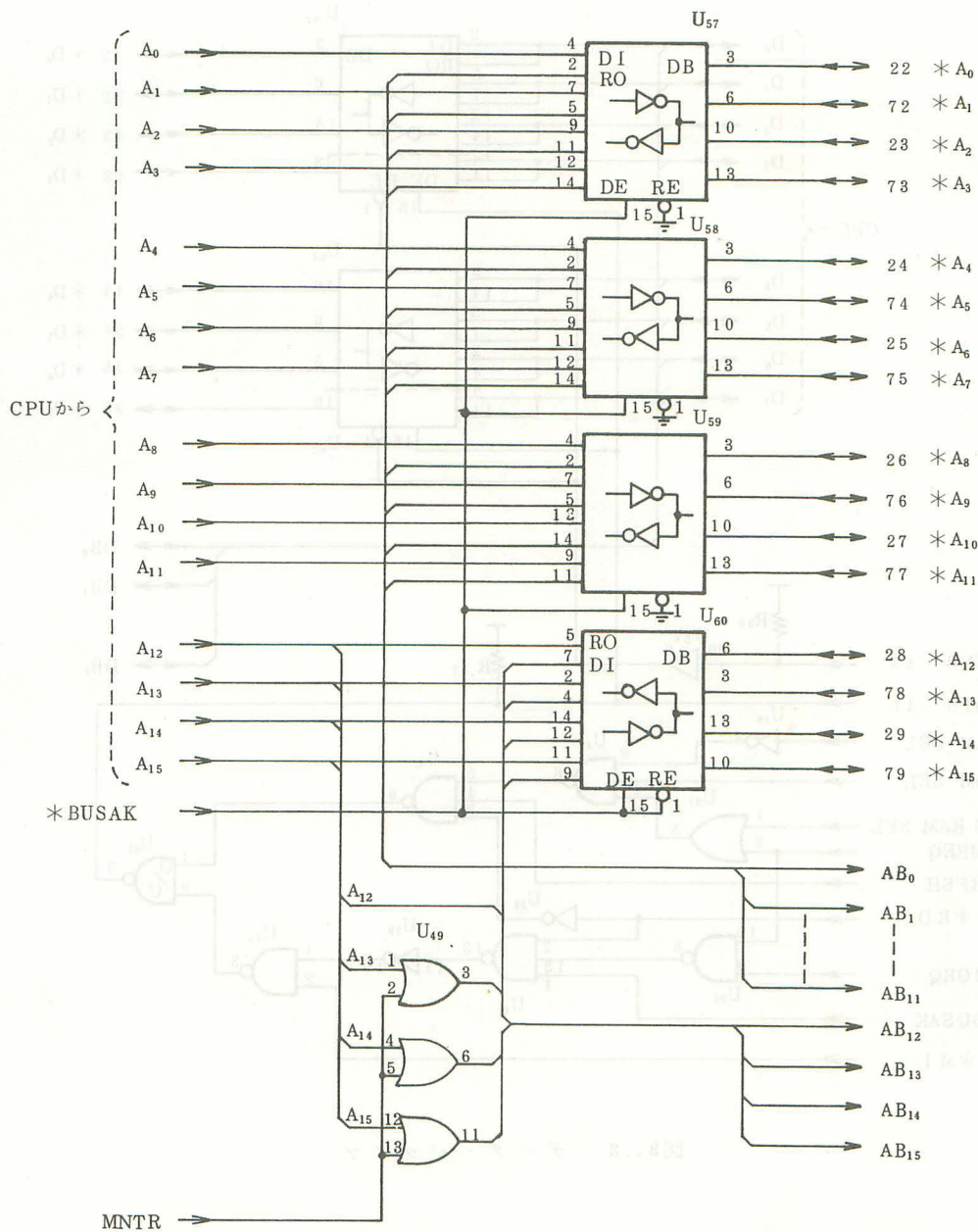
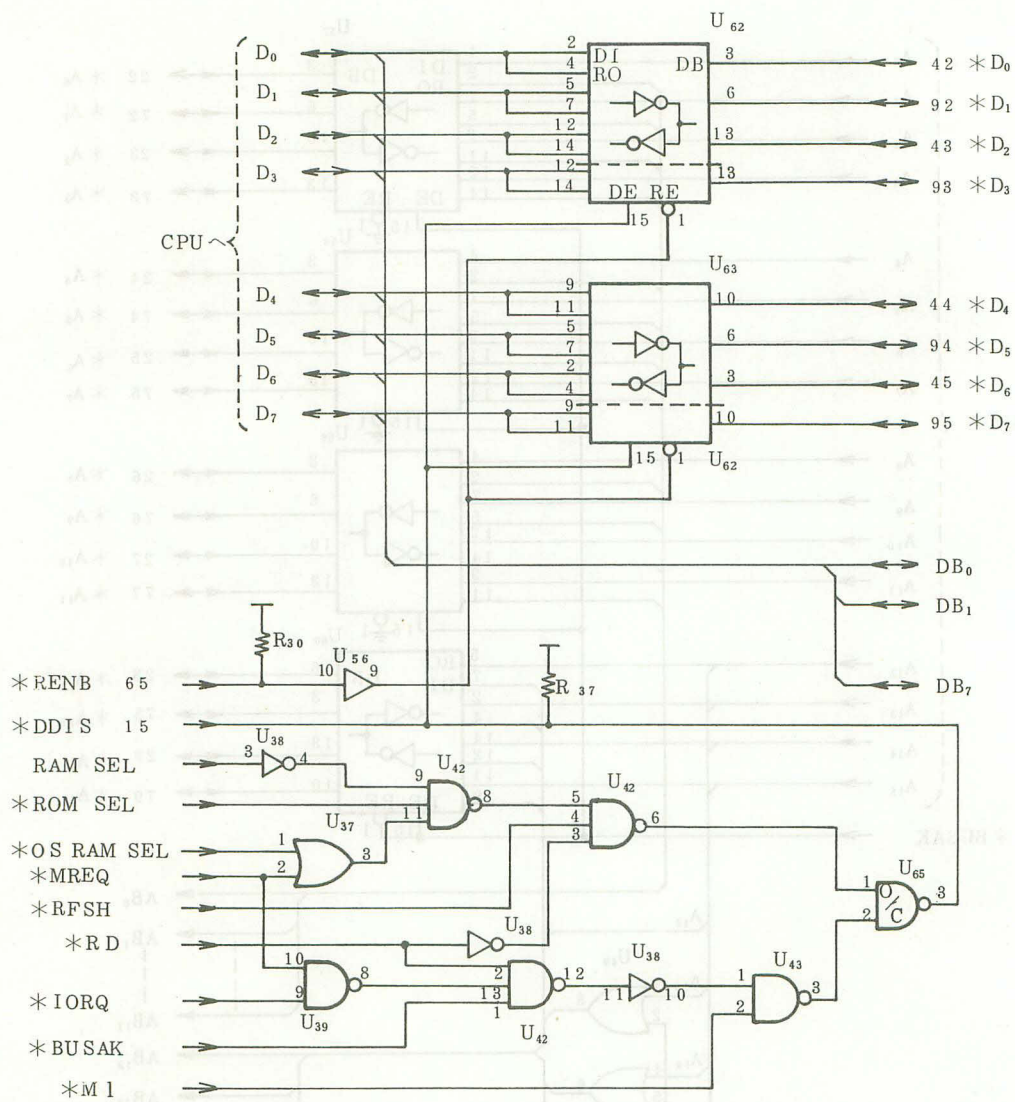



図3．3　データ・バッファ

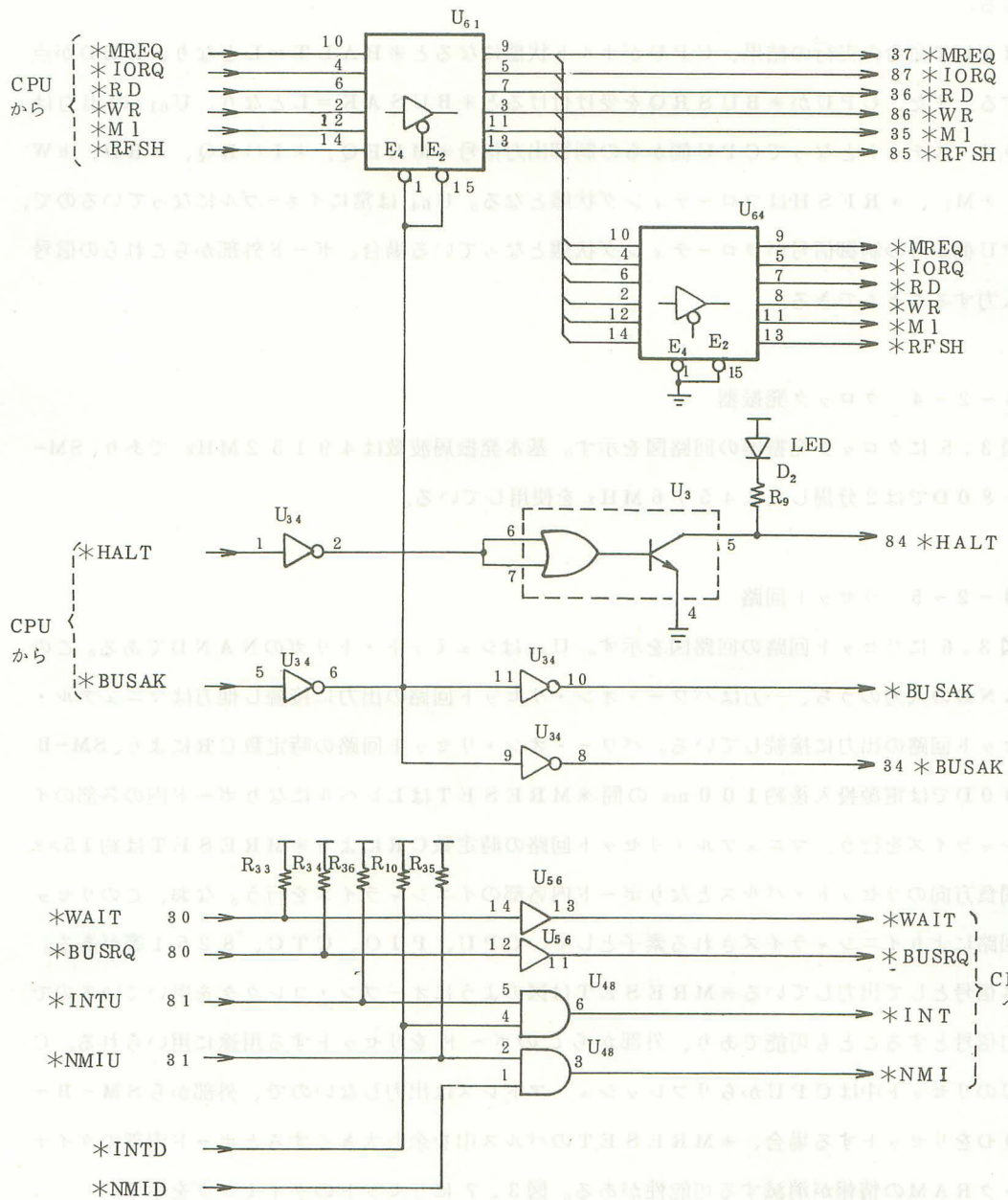



図3。4　コントロール・バッファ

である。

ＨＡＬＴ命令の実行の結果、ＣＰＵがホルト状態になると＊ＨＡＬＴ＝Ｌとなり、ＬＥＤが点灯する。また、ＣＰＵが＊ＢＵＳＲＱを受け付けると＊ＢＵＳＡＫ＝Ｌとなり、U_{61} の出力はトライ・ステートとなってＣＰＵ側からの制御出力信号＊ＭＲＥＱ、＊ＩＯＲＱ、＊ＲＤ、＊ＷＲ、＊M_1 、＊ＲＦＳＨはフローティング状態となる。U_{64} は常にイネーブルになっているので、ＣＰＵ側からの制御信号がフローティング状態となっている場合、ボード外部からこれらの信号を入力することもできる。

３－２－４　クロック発振器

図３．５にクロック発振器の回路図を示す。基本発振周波数は4.９１５２MHz であり、SM－Ｂ－８０Ｄでは２分周した2.４５７６MHz を使用している。

３－２－５　リセット回路

図３．６にリセット回路の回路図を示す。U_{44}はシュミット・トリガのＮＡＮＤである。このＮＡＮＤの入力のうち、一方はパワー・オン・リセット回路の出力に接続し他方はマニュアル・リセット回路の出力に接続している。パワー・オン・リセット回路の時定数ＣＲにより、SM－Ｂ－８０Ｄでは電源投入後約１００ms の間＊ＭＲＥＳＥＴはＬレベルになりボード内の各部のイニシャライズを行う。マニュアル・リセット回路の時定数ＣＲにより＊ＭＲＥＳＥＴは約１５μs の間負方向のリセット・パルスとなりボード内各部のイニシャライズを行う。なお、このリセット回路によりイニシャライズされる素子として、ＣＰＵ、ＰＩＯ、ＣＴＣ、８２５１等がある。バス信号として出力している＊ＭＲＥＳＥＴは図のようにオープン・コレクタを用いているので入力信号とすることも可能であり、外部からこのボードをリセットする用途に用いられる。ＣＰＵのリセット中はＣＰＵからリフレッシュ・アドレスは出力しないので、外部からＳＭ－Ｂ－８０Ｄをリセットする場合、＊ＭＲＥＳＥＴのパルス巾を余り大きくするとボード内部のダイナミックＲＡＭの情報が消滅する可能性がある。図３．７にリセットのタイミングを示す。

３－２－６　リスタート／ＮＭＩ起動回路

図３．８にリスタート／ＮＭＩ起動回路の回路図を示す。この回路の出力はＭＮＴＲと＊ＮＭＩＤである。ＭＮＴＲはアドレスの上位３ビットを強制的に１にする信号でありアドレス・バッファに入力している。＊ＮＭＩＤはモニタ・ルーチンにおいて制御されており、モニタ・コマンドのステップ、トレース、ブレイク・ポイントにおいて使用される。

図3.5　クロック発振器

VCC

C11〜C51(41個) タンタル・コンデンサ 0.22μF

D1 1S2075K

R27

C4 47μF

C47

+5V

R7

U44

U44

U65

R26

SW1

11 *MRESET

(O/C)

C3

1k

U45

U45

R38

U65

0.1μ

MRST

R8

U44

(O/C)

+5V

U45

*MRST

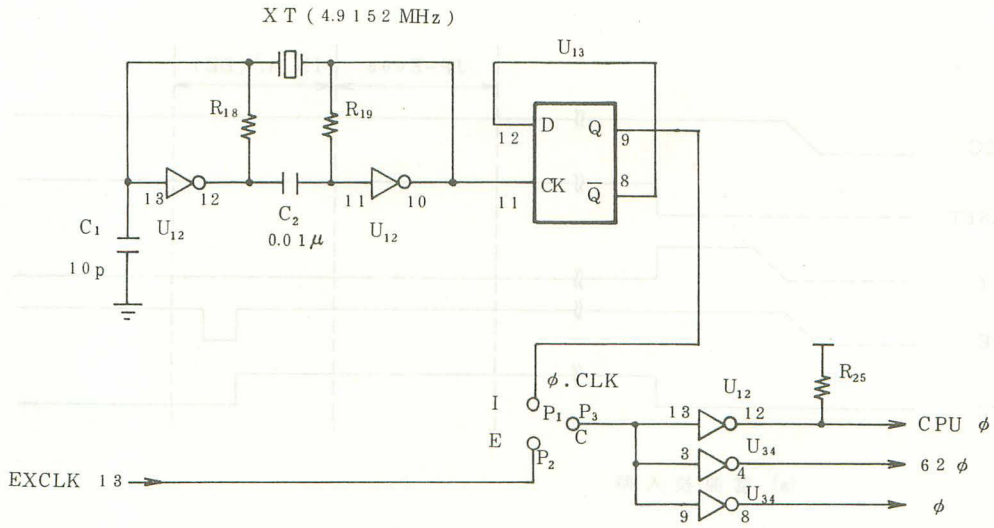
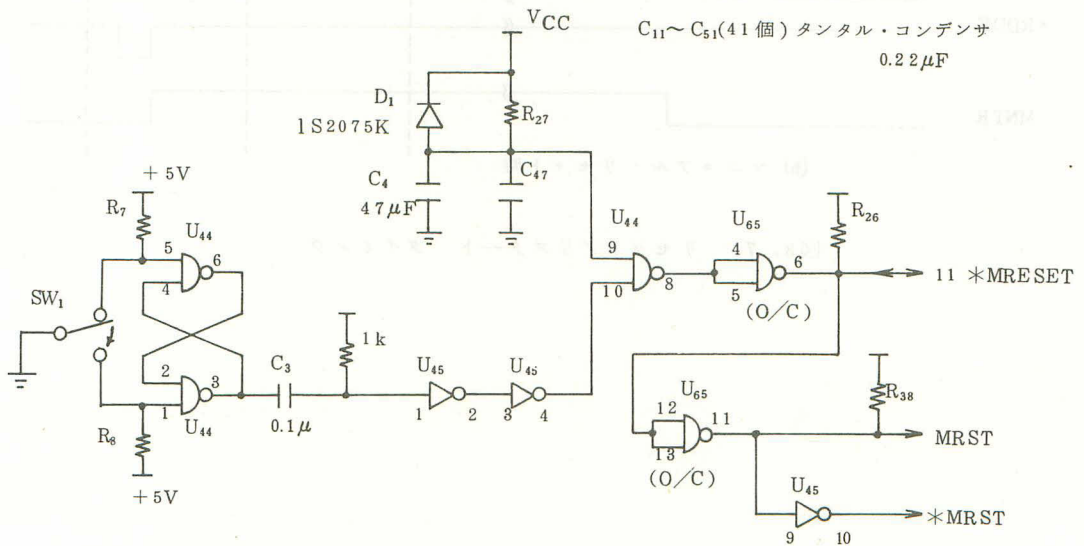



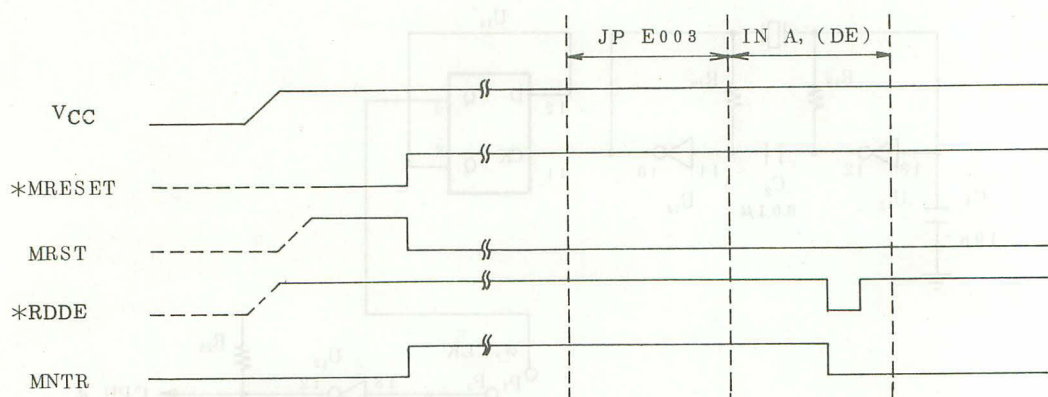
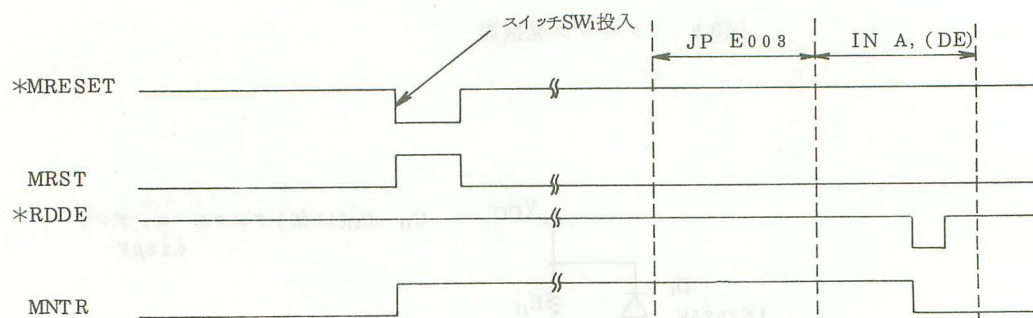
図3．6　リセット回路

(a) 電源投入時

(b) マニュアル・リセット時

図3．7　リセット／リスタート　タイミング

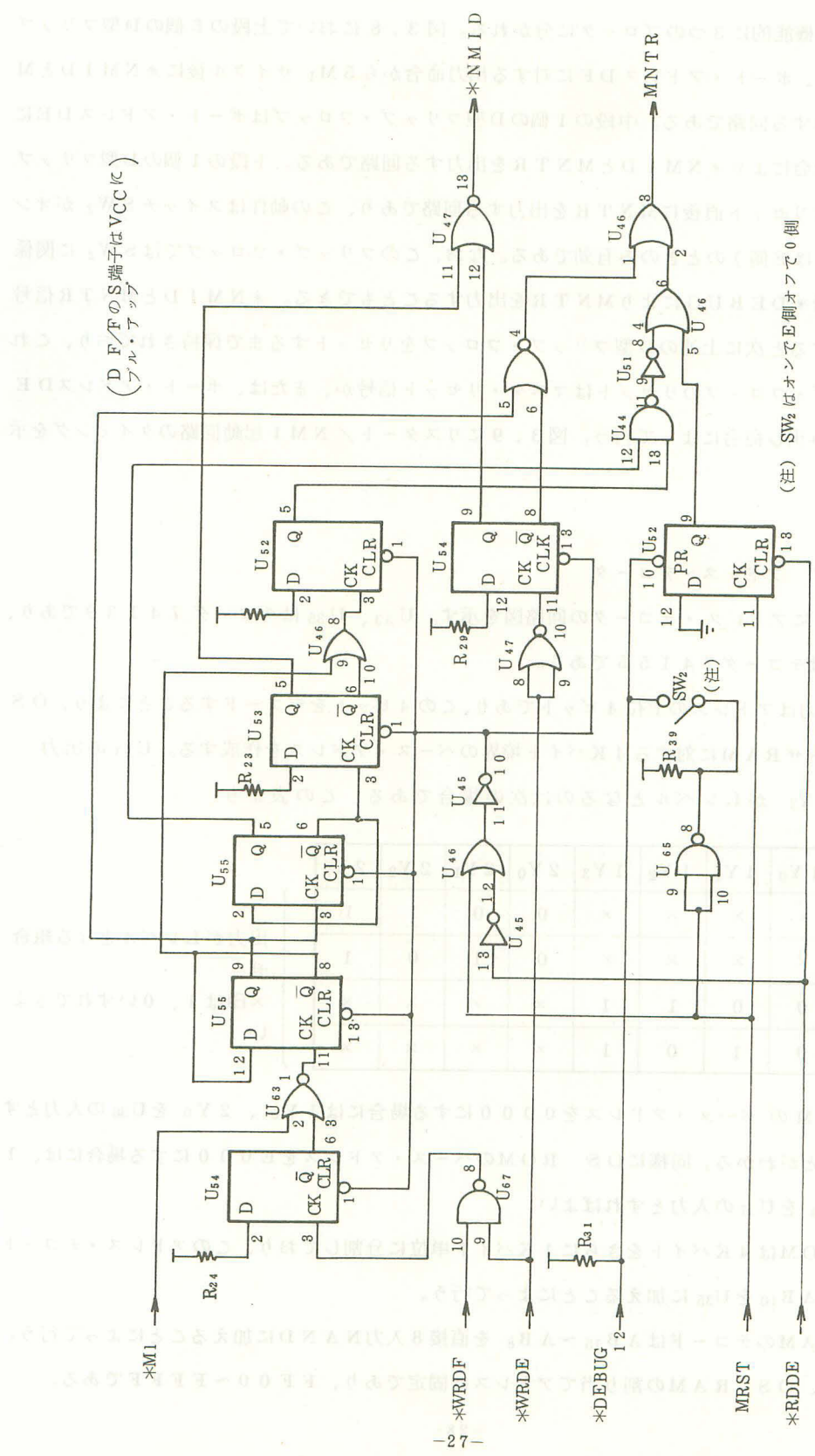



図3．8　リスタート／NMI　起動回路

この回路は機能的に３つのブロックに分かれる。図３．８において上段の５個のＤ型フリップ・フロップは、ポート・アドレスＤＦに対する出力命令から５M_1 サイクル後に＊ＮＭＩＤとＭＮＴＲを出力する回路である。中段の１個のＤ型フリップ・フロップはポート・アドレスＤＥに対する出力命令により＊ＮＭＩＤとＭＮＴＲを出力する回路である。下段の１個のＤ型フリップ・フロップはリセット直後にＭＮＴＲを出力する回路であり、この動作はスイッチSW_2 がオン（ボード上ではＥ側）のときのみ有効である。なお、このフリップ・フロップではSW_2 に関係なく外部信号＊ＤＥＢＵＧによりＭＮＴＲを出力することもできる。＊ＮＭＩＤとＭＮＴＲ信号は一度出力すると次に上述のＤ型フリップ・フロップをリセットするまで保持されており、これらのフリップ・フロップのリセットはマスタ・リセット信号か、または、ポート・アドレスＤＥに対する読み出し命令によって行う。図３．９にリスタート／ＮＭＩ起動回路のタイミングを示す。

３－２－７　アドレス・デコーダ

図３．１０ にアドレス・デコーダの回路図を示す。U_{33}、U_{35} はデコーダ７４１３９であり、U_4 、U_6 はデコーダ７４１５５である。

U_{33} の入力はアドレスの上位４ビットであり、この４ビットをデコードすることにより、ＯＳ　ＲＯＭ、ユーザＲＡＭに対する４Ｋバイト境界のベース・アドレスを作成する。U_{33} の出力 $1Y_0$ ～ $2Y_3$ がＬレベルとなるのは次の場合である。この表より

	$1Y_0$	$1Y_1$	$1Y_2$	$1Y_3$	$2Y_0$	$2Y_1$	$2Y_2$	$2Y_3$
AB_{15}	×	×	×	×	0	0	1	1
AB_{14}	×	×	×	×	0	1	0	1
AB_{13}	0	0	1	1	×	×	×	×
AB_{12}	0	1	0	1	×	×	×	×

出力がＬレベルとなる組合せ
×印は１、０いずれでもよい

ユーザＲＡＭのベース・アドレスを００００にする場合には$1Y_0$ 、$2Y_0$ をU_{36} の入力とすればよいことがわかる。同様にＯＳ　ＲＯＭのベース・アドレスをＥ０００にする場合には、$1Y_2$ 、$2Y_3$ をU_{37}の入力とすればよい。

ＯＳ　ＲＯＭは４Ｋバイトをさらに１Ｋバイト単位に分割しており、このアドレス・デコードはAB_{11}、AB_{10} をU_{35} に加えることによって行う。

ＯＳ　ＲＡＭのデコードはAB_{15}～AB_8 を直接８入力ＮＡＮＤに加えることによって行う。したがって、ＯＳ　ＲＡＭの割り当てアドレスは固定であり、ＦＦ００～ＦＦＦＦである。

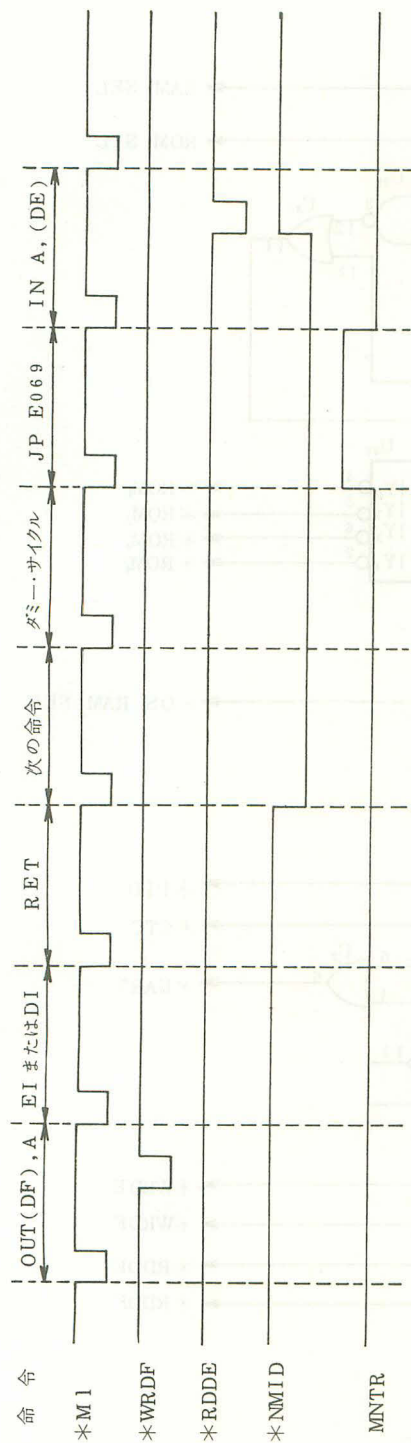
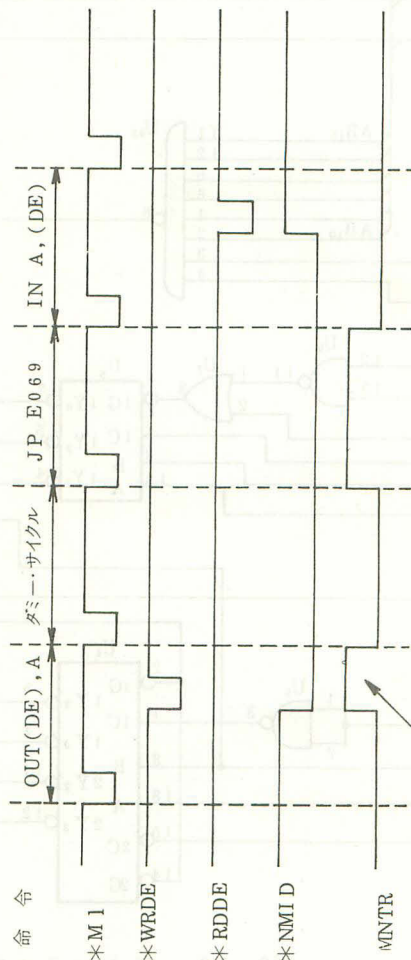



図3．9　リスタート／NMI　起動回路タイミング

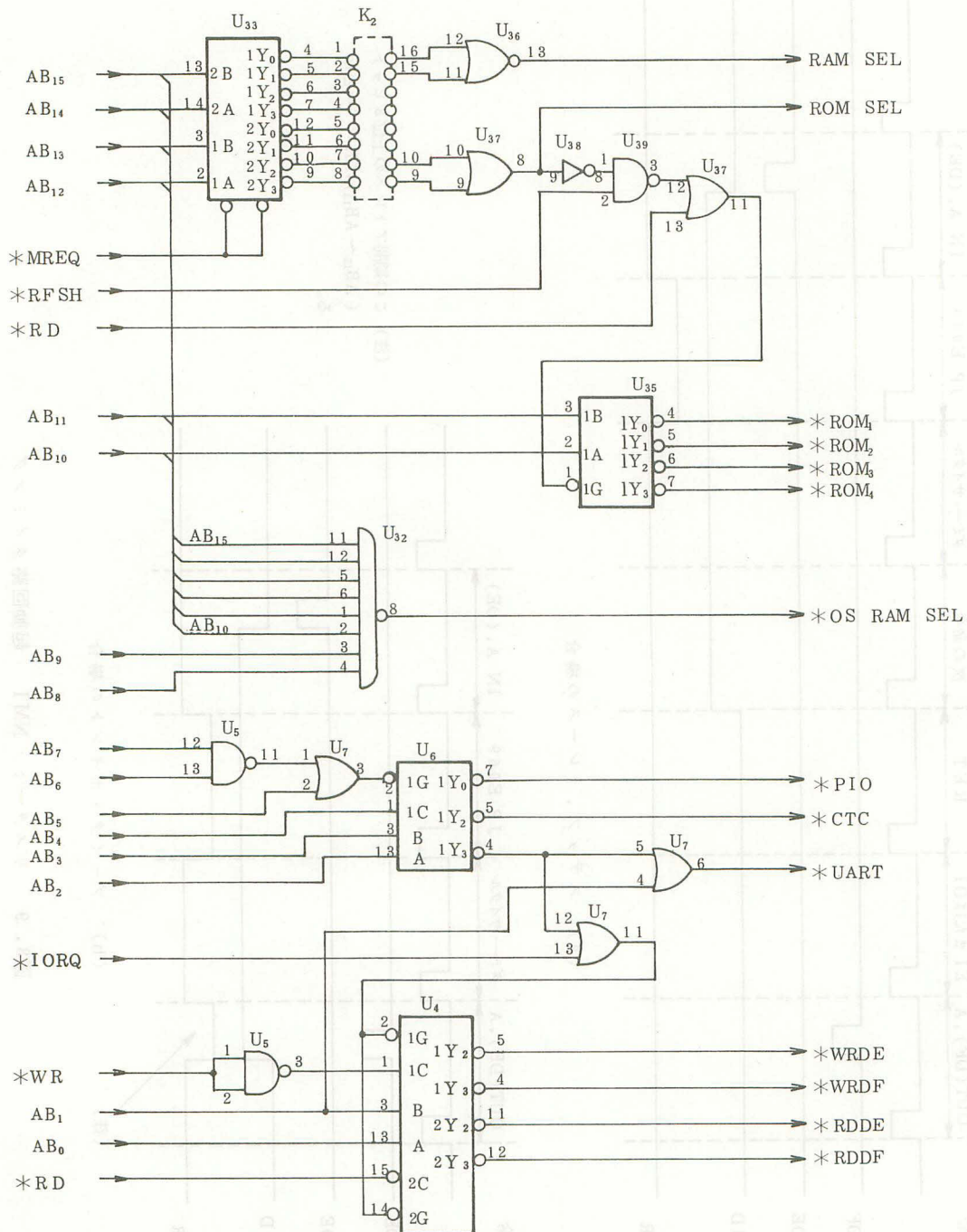



図3．10　アドレス・デコーダ

U_4 、U_6 はポート・アドレスのデコーダであり、各出力信号がLレベルになる条件を次の表に示す。 表で×印の個所は0、1いずれでもよいことを示しているが、PIO、 CTC、UARTを選択する場合、これらの×印の信号は各デバイス側では確定している。

	＊PIO	＊CTC	＊UART	＊WRDE	＊WRDF	＊RDDE	＊RDDF
AB_7	1	1	1	1	1	1	1
AB_6	1	1	1	1	1	1	1
AB_5	0	0	0	0	0	0	0
AB_4	1	1	1	1	1	1	1
AB_3	0	1	1	1	1	1	1
AB_2	0	0	1	1	1	1	1
AB_1	×	×	0	1	1	1	1
AB_0	×	×	×	0	1	0	1
＊IORQ	×	×	×	0	0	0	0
＊RD	×	×	×	1	1	0	0
＊WR	×	×	×	0	0	1	1

3－2－8 OS ROM

OS ROMはソケットにより実装し、2708タイプのEPROMを最大4個まで装着できる。各EPROMは、アドレス・デコーダの出力信号＊ROM_1 ～ROM_4 によって選択される。OS ROMとしてEPROMの代りに、2708とピン互換性のあるバイポーラPROMを実装することもできるが、この場合、K_3 によって電源線を一部変更しなければならない。（表3.10参照） バイポーラPROMの例 LH－7055。

3－2－9 OS RAM

OS RAMとして2111タイプのスタティックRAMを2個使用している。このタイプのメモリは256×4ビットの構成である。OS RAMの選択信号である＊OS RAM SEL はアドレスFF00～FFFFの範囲でLレベルになり、これをメモリの＊CE_1 端子に印加している。また、メモリの＊CE_2 端子には＊MREQ、R/W端子には＊WR、OD端子には＊RD信号を印加している。

+12V
−5V
+5V
K 3
AB0
AB 1
AB 9
AB9
AB0
+5V
A9
A0
VCC
CS/WE
VDD
1
VBB
PRG
VSS
O7
O0
U19
U20
U21
*ROM1
*ROM2
*ROM3
*ROM4
4
U22
DB0
DB1
DB7

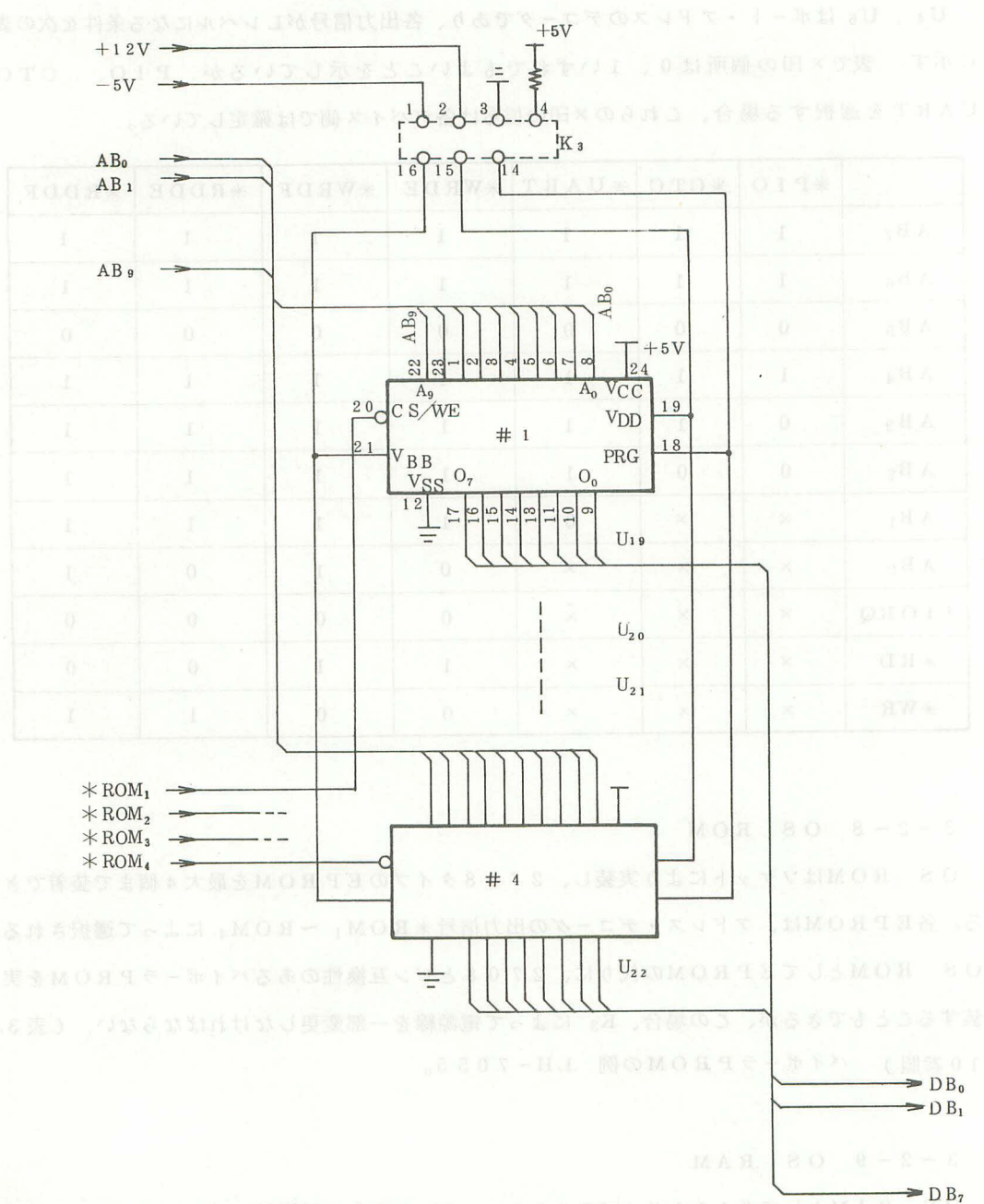
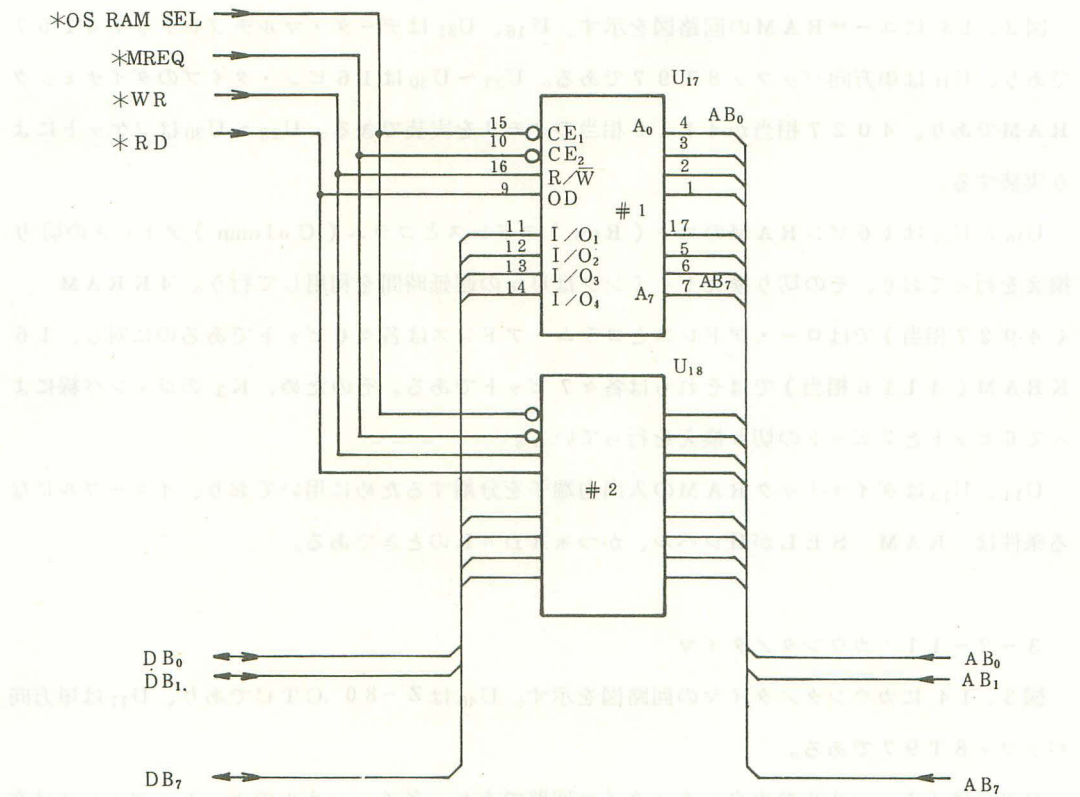



図3．11　　OS　ROM

3－2－10　ユーザRAM

図3．13 にユーザRAMの回路図を示す。U_{16}、U_{31} はデータ・マルチプレクサ74157であり、U_{14} は単方向バッファ8T97である。U_{23} ～ U_{30} は16ピン・タイプのダイナミックRAMであり、4027相当か4116相当のメモリを実装できる。U_{23} ～ U_{30} はソケットにより実装する。

U_{16} と U_{31} は16ピンRAMのロー（Row）アドレスとコラム（Column）アドレスの切り換えを行っており、その切り換えタイミングは U_{50} の遅延時間を利用して行う。4KRAM（4027相当）ではロー・アドレスとコラム・アドレスは各々6ビットであるのに対し、16KRAM（4116相当）ではそれらは各々7ビットである。そのため、K_3 のジャンパ線によって6ビットと7ビットの切り換えを行っている。

U_{14}、U_{15} はダイナミックRAMの入出力端子を分離するために用いており、イネーブルになる条件は　RAM　SELがHレベル、かつ＊RD＝Lのときである。

3－2－11　カウンタ／タイマ

図3．14 にカウンタ／タイマの回路図を示す。U_{40} はZ－80　CTCであり、U_{11} は単方向バッファ8T97である。

CTCは4チャンネルのカウンタ／タイマ回路であり、各チャンネルのポート・アドレスは次のようになっている。3.2.8の表の＊CTCの項参照。

チャンネル0	D8
チャンネル1	D9
チャンネル2	DA
チャンネル3	DB

通常チャンネル0はボー・レート・クロック作成用としてモニタで使用しており、ユーザ側で使用してはならない。チャンネル0の出力ZC／TO_0 はボー・レートの16倍のクロックが出力しており、これを UART のシリアル・クロックとしている。シリアル・クロックをXSCLK端子を用いて外部から供給する場合、チャンネル0をユーザ側で使用してもよい。チャンネル1～チャンネル3はユーザに開放されている。チャンネル1の出力ZC／TO1はチャンネル0の入力CK／TRG0に接続しているが、これはチャンネル0、チャンネル1を縦続接続にすることによってカウント数を大きくするためのものである。

モニタではチャンネル0をタイマ・モードで使用しており、基本クロックはCPUに供給するクロックと同じ周波数2.4576MHz としている。シリアル・クロック（SER　CLK）は

*WR
U31
U23
AB7
AB0
AB8
AB1
AB9
AB2
AB10
AB3
1A
1B
2A
2B
3A
3B
4A
4B
1Y
2Y
3Y
4Y
E
S
W
A0
A1
A2
A3
A4
A5
A6
Di
#1
D0
CS
CAS RAS
U16
AB11
AB4
K3
AB5
AB12
AB6
AB13
U24
R20
R21
#2
U25
~
U29
U50
U36
U37
U38
U39
U30
*MREQ
#7
*RFSH
RAM SEL
*RD
U14
E4
DB0
DB1
DB2
DB3
U15
E2
DB4
DB5
DB6
DB7

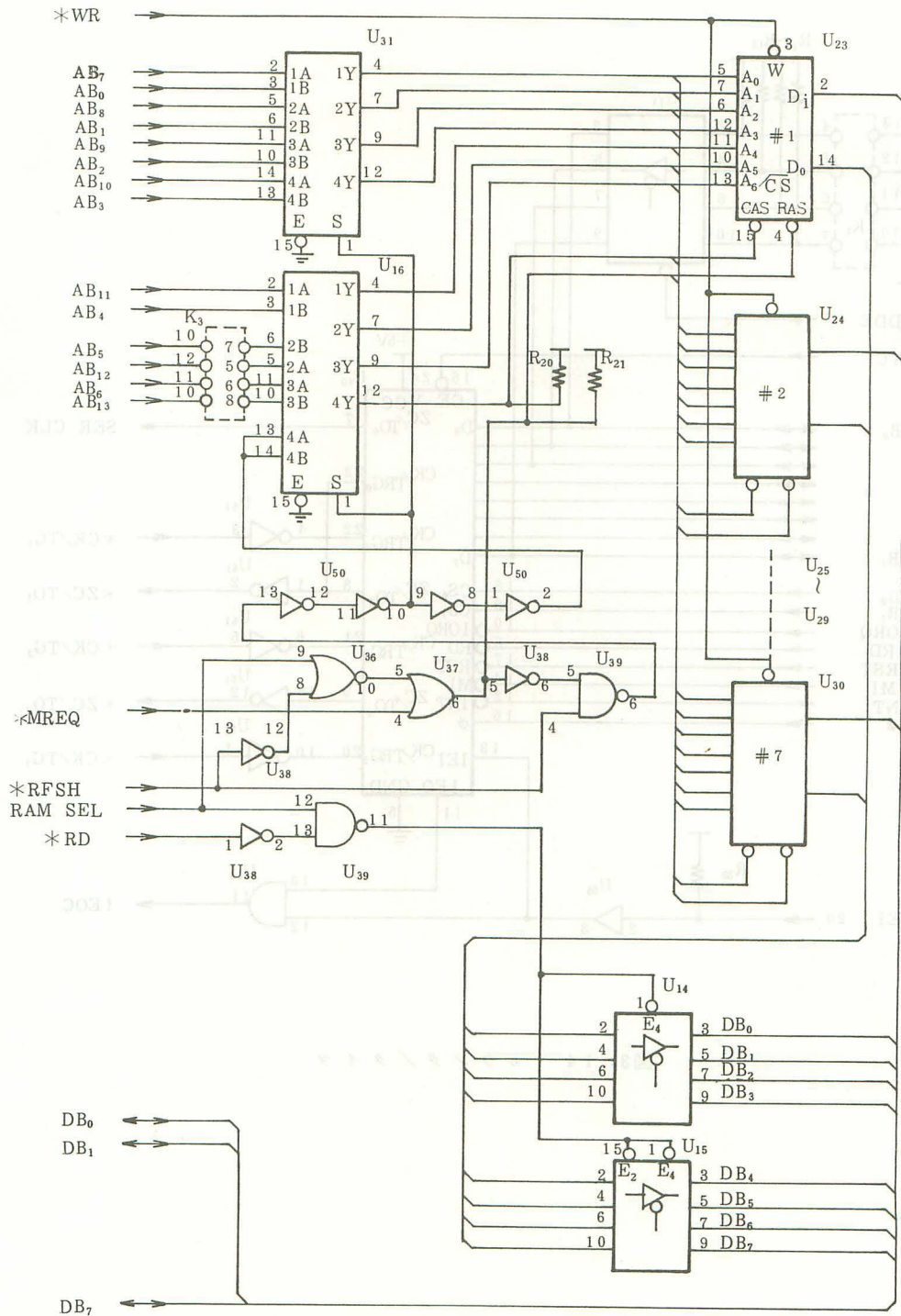



図3．13　ユーザ RAM

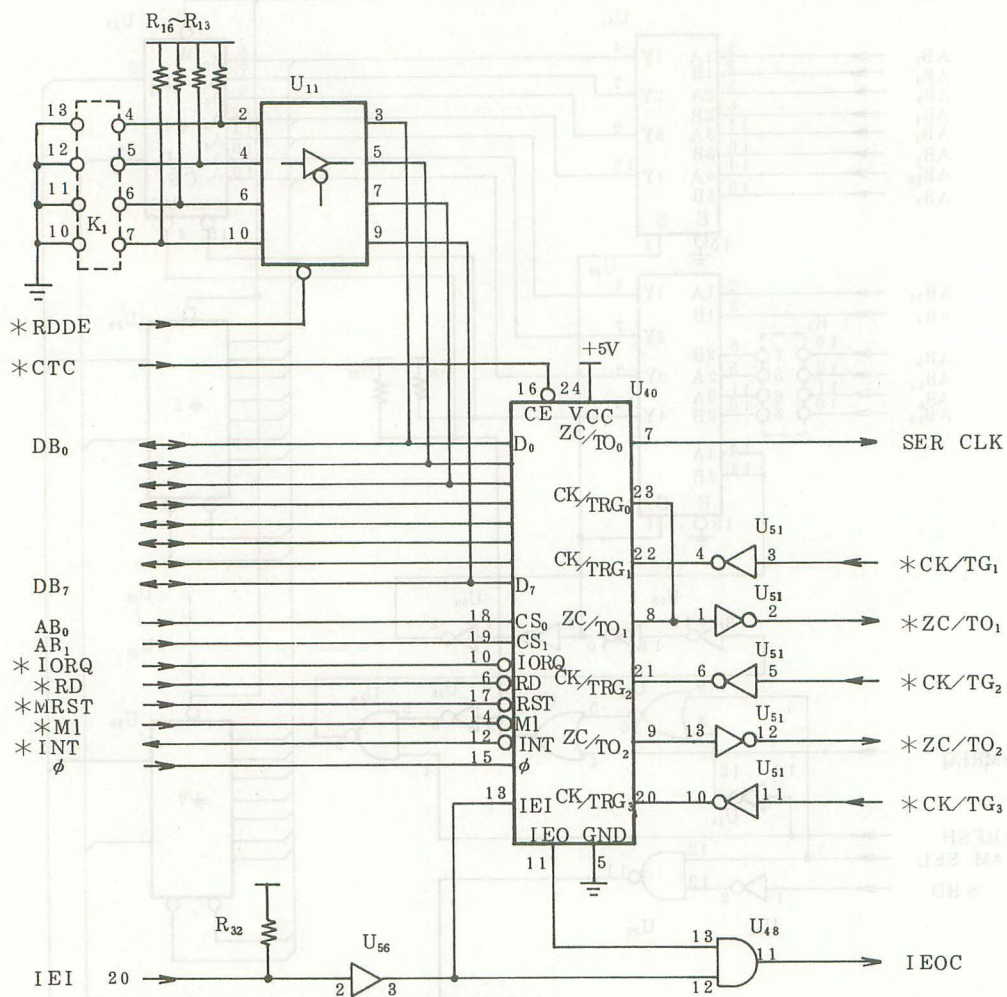
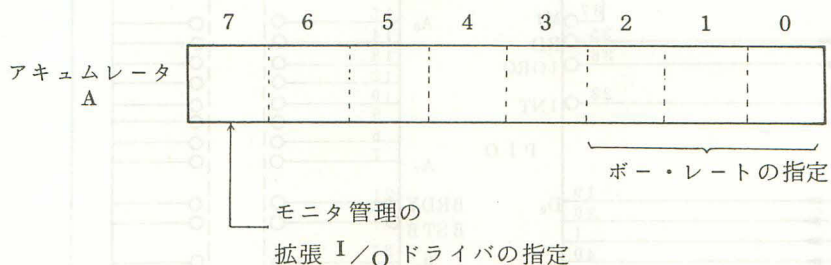



図3．14　カウンタ／タイマ

ボー・レートに応じてこのクロックを分周することによって得ている。

図のK_1、U_{11}はボー・レートとモニタ管理の拡張 I/O ドライバの指定に用いるものである。プログラム・ジャンパK_1の状態はポート・アドレスDEに対する入力命令によりアキュムレータAに読み込まれる。この場合、アキュムレータAの内容は次の意味を持っている。ここでモニタ管理の拡張 I/O ドライバとは、SM−B−80 Dのデータの入出力形式であるCI、CO、SI、SO、OI、OOに対する I/O 装置もユーザ側で定義した場合を指す。これらユーザ拡張

の I/O 装置のドライバ・ルーチンをOS ROMの＃2、＃3に置き、上図アキュムレータAのビット7を0にすることにより、 I/O 装置をモニタ管理下に置くことができる。この動作については、4.2.2 I/O チャンネルの項において説明する。（K_1の端子7−10を接続すると、ポートDEに対する入力命令を実行するとアキュムレータのビット7は0になる。）

3−2−12　パラレル I/O インターフェース

図3.15 はパラレル I/O インターフェースの回路図である。U_{51}はZ80　PIOである。

PIOは汎用の I/O インターフェース用LSIであり、8ビットの I/O ポートと2本のハンド・シェーク制御線を2個（Aポート、Bポートと呼ぶ）有している。SM−B−80 DではAポート、Bポートに配線はしておらず、 I/O 装置とPIO間のインターフェース回路はユーザ側に委せられている。ユーザ領域には16ピンDIP　ICを4個まで置くことができる。

PIOには4個の8ビット・レジスタがあるが、これらに対するアドレスは次のように決まっている。3.2.8の表の＊PIOの項参照。

PIO	Aポート	データ・レジスタ	D_0
	Aポート	コントロール・レジスタ	D_1
	Bポート	データ・レジスタ	D_2
	Bポート	コントロール・レジスタ	D_3

PIOのM_1端子には＊MRSTと＊M_1のAND信号を印加しているが、これはM_1端子によりPIOのイニシャライズを行うためである。

図3．15 パラレル I/O インターフェース

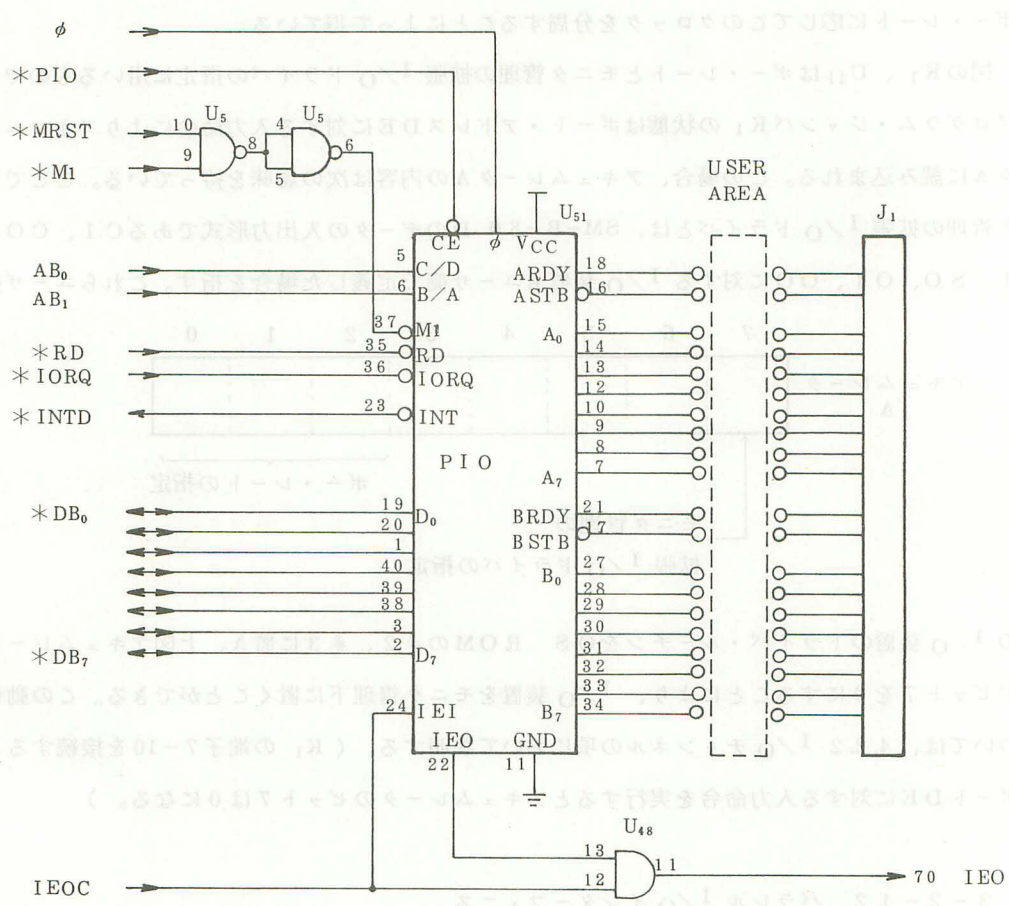
図3．14、図3．15のU_{48}、U_{56}によりCTC、PIOのデージー・チェーンを行っている。PIOの各入出力信号線については、"Z－80－PIO テクニカル マニュアル"を参照のこと。

3－2－13 シリアル I/O インターフェース

図3．16にシリアル I/O インターフェースの回路図を示す。U_{10}はUSART（Universal Synchronous Asynchronous Receiver Transmitter）8251であり、U_8はライン・ドライバ75188、U_9はライン・レシーバ75189である。

8251はパラレル・データとシリアル・データの変換機能を有しているが、その動作についての説明は省略する。シリアル I/O 装置としては、20mA電流ループのTTYとRS－232C規模の I/O 装置が接続できるようになっている。SM－B－80 DではTTY用としてU_2、U_9により電流ドライブのインターフェースを構成しており、RS－232C規格の装置用としてU_8、U_9により電圧ドライブのインターフェースを構成している。

8251のステータス・レジスタにおいて、Tx E、Rx RDY、Tx RDYビットがセットされると対応する出力端子TE、RR、TRがHレベルになる。これらのステータス・ビットによりCPUに割り込みをかける必要がある場合、図に示すK_1にジャンパ線を接続すればよい。

8251に対するポート・アドレスは次のように決まっている。

UART	データ・レジスタ	DC（注）
	コマンド・レジスタ	DD

U_{13}はリーダ制御用のフリップ・フロップであり、この出力はポート・アドレスDFに対する入力命令によりセットされ、シリアル・データのLレベル（リーダから読み込まれたデータのスタート・ビット）によってリセットする。

シリアル I/O インターフェースはフラット・ケーブル用コネクタJ_2を介して行う。各インターフェース信号線の信号名、端子番号、信号説明を次の表にまとめる。（注）本マニュアルではUSARTをUARTと略している。

3－2－14 その他

SM－B－80 Dのその他の回路を図3．17、図3．18に示す。図3．17はZ－80 CPUの各信号線である。図3．18はボードの電源回路図である。

図3．19はSM－B－80 Dの部品実装図であり、表3.1は対応する部品構成リストである。

図3．16　シリアル I/O インターフェース

図3．17　Z－80　CPU　信号線

1, 2, 3, 51, 52, 53
+5V
C5
C
+5V
5, 55
+12V
C6
C
+12V
46, 96
−12V
C7
R40
C
−12V
U66
IN
OUT
CMN
−5V
C8
C9
C10
C
48, 49, 50
98, 99, 100
GND
GND
（注） C：C11～C51
（パスコン，0.22μF）

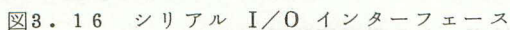
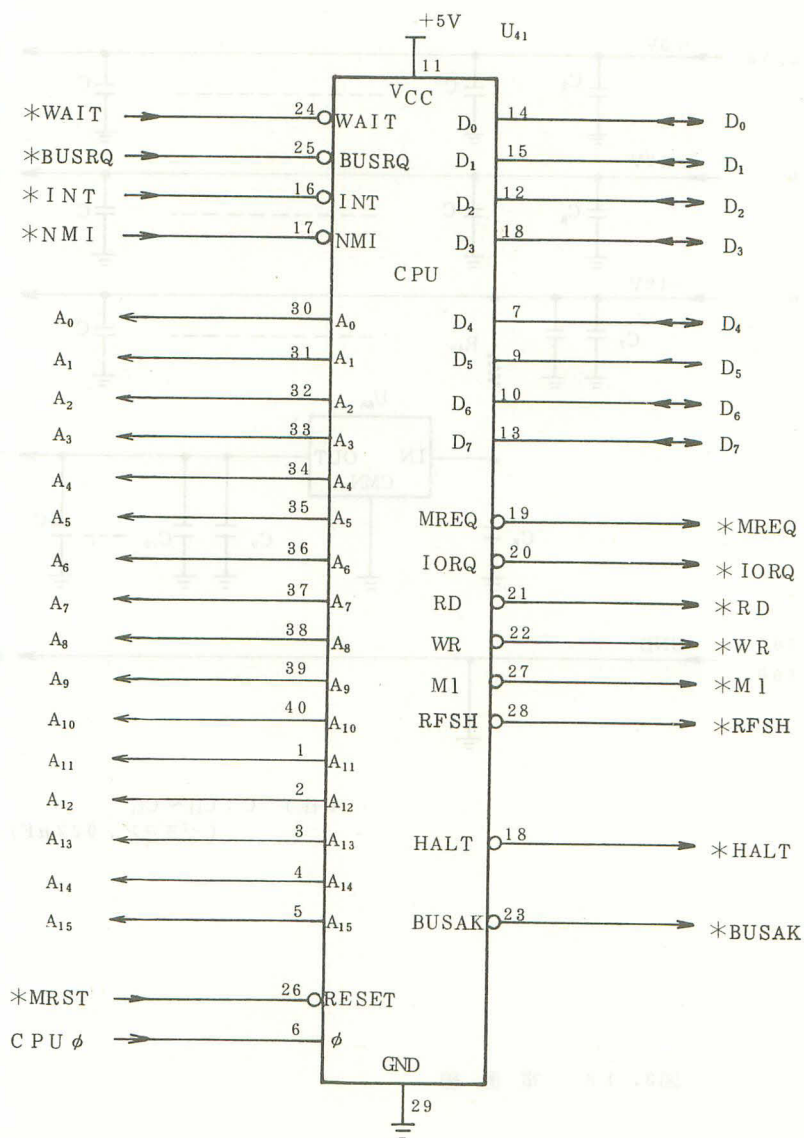
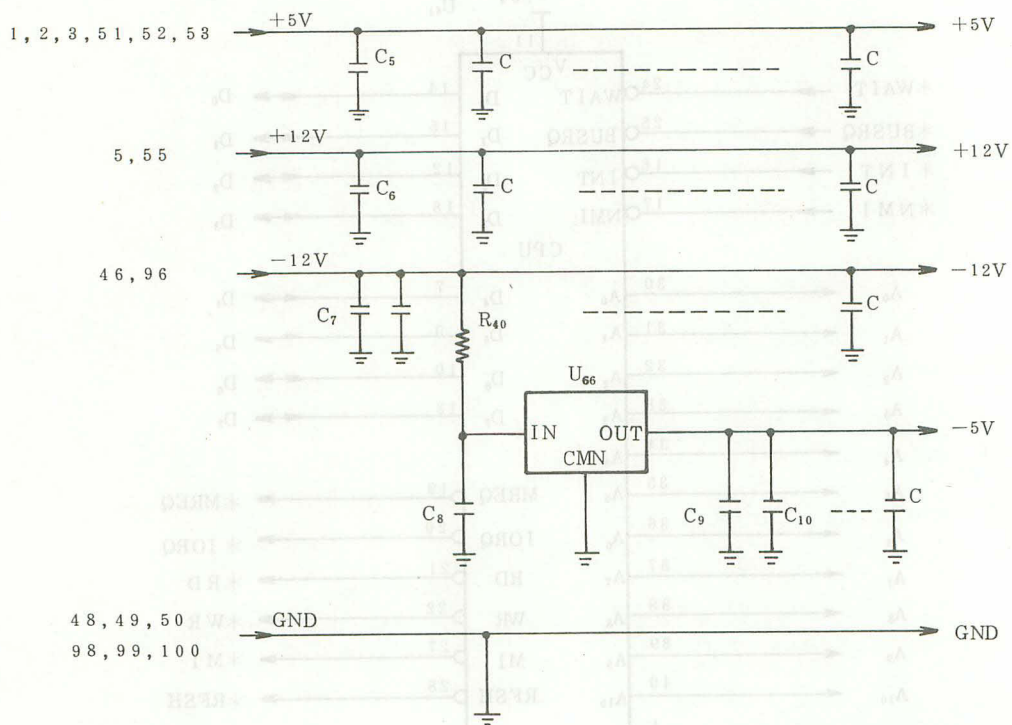



図3．18　電 源 線

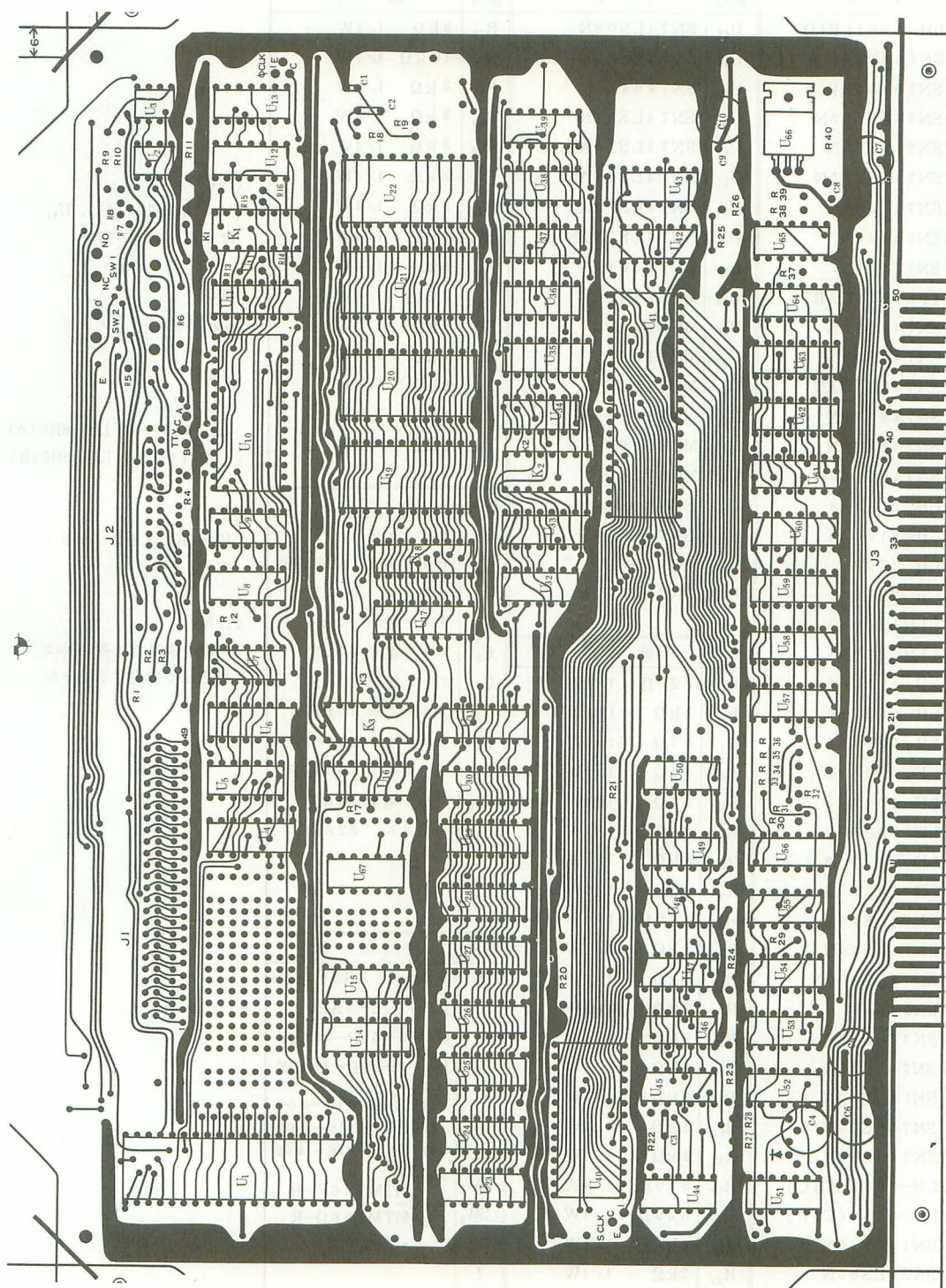



図3.19　SM－B－80D　部品実装図

表3．1　SM－B－80D　主要部品構成リスト

番号	I C	番号	I C	番号	抵抗
U_{1}	LH－0081(PIO)	U_{48}	SN74LS08N	R_{26}	3kΩ　1/4W
U_{2}	SN75454BP	U_{49}	SN74LS32N	R_{27}	10kΩ　1/4W
U_{3}	SN75454BP	U_{50}	SN7404N	R_{28}	3kΩ　1/4W
U_{4}	SN74LS155N	U_{51}	SN74LS04N	R_{29}	3kΩ　1/4W
U_{5}	SN74LS00N	U_{52}	SN74LS74AN	R_{30}	3kΩ　1/4W
U_{6}	SN74LS155N	U_{53}	SN74LS74AN	R_{31}	3kΩ　1/4W
U_{7}	SN74LS32N	U_{54}	SN74LS74AN	R_{32}	3kΩ　1/4W
U_{8}	SN75188N	U_{55}	SN74LS74AN	R_{33}	3kΩ　1/4W
U_{9}	SN75189N	U_{56}	SN74367AN	R_{34}	3kΩ　1/4W
U_{10}	iP8251(UART)	U_{57}	MC8T26P	R_{35}	3kΩ　1/4W
U_{11}	SN74367AN	U_{58}	MC8T26P	R_{36}	3kΩ　1/4W
U_{12}	SN74LS04N	U_{59}	MC8T26P	R_{37}	1kΩ　1/4W
U_{13}	SN74LS74AN	U_{60}	MC8T26P	R_{38}	3kΩ　1/4W
U_{14}	SN74367AN	U_{61}	SN74367AN	R_{39}	3kΩ　1/4W
U_{15}	SN74367AN	U_{62}	MC8T26P	R_{40}	22Ω　1W
U_{16}	SN74LS157N	U_{63}	MC8T26P		
U_{17}	LH－2111A4	U_{64}	SN74367AN	番号	コンデンサ
U_{18}	LH－2111A4	U_{65}	SN74LS03N	C_{1}	セラミック 10PF
U_{19}	LH－2708	U_{66}	MC7905CP	C_{2}	セラミック 10000PF
U_{20}	LH－2708	U_{67}	SN74LS00N	C_{3}	タンタル　0.1μF
U_{21}	(LH－2708)			C_{4}	アルミ電解 47μF
U_{22}	(LH－2708)	番号	抵抗	C_{5}	アルミ電解 47μF
U_{23}	LH－4027－3	R_{1}	220Ω　1/4W	C_{6}	アルミ電解 47μF
U_{24}	LH－4027－3	R_{2}	3kΩ　1/4W	C_{7}	アルミ電解 4.7μF
U_{25}	LH－4027－3	R_{3}	47Ω　1/4W	C_{8}	セラミック 1000PF
U_{26}	LH－4027－3	R_{4}	47Ω　1/4W	C_{9}	セラミック 1000PF
U_{27}	LH－4027－3	R_{5}	47Ω　1/4W	C_{10}	アルミ電解 47μF
U_{28}	LH－4027－3	R_{6}	220Ω　1/4W	C_{11}	タンタル　0.22μF
U_{29}	LH－4027－3	R_{7}	3kΩ　1/4W	≀	≀
U_{30}	LH－4027－3	R_{8}	3kΩ　1/4W	C_{51}	〃　0.22μF
U_{31}	SN74LS157	R_{9}	220Ω　1/4W		
U_{32}	SN74LS30N	R_{10}	3kΩ　1/4W	番号	その他
U_{33}	SN74LS139N	R_{11}	3kΩ　1/4W	D_{1}	ダイオード 1S2075K
U_{34}	SN7404N	R_{12}	3kΩ　1/4W	D_{2}	LED GL－5AR1
U_{35}	SN74LS139N	R_{13}	3kΩ　1/4W	J_{1}	H1F3－50P－2.54DS
U_{36}	SN74LS02N	R_{14}	3kΩ　1/4W	J_{2}	H1F3－26P－2.54DS
U_{37}	SN74LS32N	R_{15}	3kΩ　1/4W	K_{1}	フラット・ホーム DIS02－016－402
U_{38}	SN74LS04N	R_{16}	3kΩ　1/4W	K_{2}	DIS02－016－402
U_{39}	SN74LS00N	R_{17}	3kΩ　1/4W	K_{3}	DIS02－016－402
U_{40}	LH－0082(CTC)	R_{18}	820Ω　1/4W	SW_{1}	スイッチ MTM106F－R
U_{41}	LH－0080(CPU)	R_{19}	820Ω　1/4W	SW_{2}	スイッチ MTM106D－R
U_{42}	SN74LS10N	R_{20}	2kΩ　1/4W	P_{1}	チェック端子
U_{43}	SN74LS00N	R_{21}	2kΩ　1/4W	≀	
U_{44}	SN74LS132N	R_{22}	1kΩ　1/4W	P_{9}	チェック端子
U_{45}	SN7404N	R_{23}	3kΩ　1/4W	XT	水晶振動子 4.9152MHz
U_{46}	SN74LS32N	R_{24}	3kΩ　1/4W		
U_{47}	SN74LS02N	R_{25}	330Ω　1/4W		

(注1)
U_{1}, U_{19}〜U_{30}, U_{40}, U_{41} はICソケット実装

(注2)
U_{21}, U_{22} はオプション

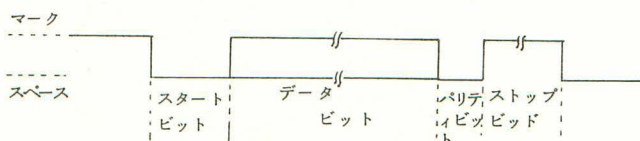
(注3)
U_{23}〜U_{30}
LH－4027－3(LH－8H01A)
LH－4116－3(LH－8H01B)

(注4)
K_{1}〜K_{3} はICソケット実装

(注5)
実際の製品では各部品は相当品に変更する場合がある。

シリアル I/O インターフェイス　信号線説明

信号名	J_2番号	信号説明
GND	1 13	接地線
RS(+) RS(−)	6 16	リーダ・ステップ　リーダ・オン時に　RS(+)、RS(−)を通じて電流が流れる。(＋12Vで直列抵抗94Ω)
CARRIER DETECT	15	20mA SEND信号と同一信号。20mA SEND端子開放時は本端子は＋12Vである。電流ループ使用の場合、UARTの送信データありの状態(スペース)でHレベル、なしの状態(マーク)でLレベルとなる。(注1)
20mA SEND 20mA SEND RET	24 25	20mA電流ループでTTY(ASR−33)を使用する場合、本信号を用いる。UARTの送信データがマークの状態でループ電流オン。スペースの状態でループ電流オフとなる。
20mA REC 20mA REC RET	22 23	20mA電流ループでTTY(ASR−33)を使用する場合　本信号を用いる。TTYからの送信データがマークの状態で端子22はLレベルに、スペースの状態でHレベルになっている。
TRANS DATA	3	RS−232C規格で、端末装置からデータが送られてくる場合に用いる入力端子。受信データがマークの状態でLレベルに、スペースの状態でHレベルになっている。
REC DATA	5	RS−232C規格で端末装置へデータを送る場合に用いる出力端子。送信データがマークの状態でLレベルに、スペースの状態でHレベルになっている。
DSR (Data Set Ready)	11	UARTの$\overline{DTR}$信号の反転信号。RS−232C規格。出力。 UART(8251)のコマンド　ビット1を　セットすると$\overline{DTR}$=0となる。
CTS (Clear To Send)	9	UARTの$\overline{RTS}$信号の反転信号。RS−232C規格。出力。 UARTのコマンド　ビット5をセットすると$\overline{RTS}$=0となる。
DTR (Data Term Ready)	14	本信号の反転信号がUARTの$\overline{DSR}$に等しい。RS−232C規格。入力。UARTの$\overline{DSR}$をLレベルにするとステータス　ビット7がセットされる。
RTS (Request To Send)	7	ボード上のチェック端子(記号TTY)の結線状態により動作が異なる。 TTY　A−C結線；RTS信号　無効 A−B結線；RTSの反転信号＝UARTの$\overline{CTS}$ 本信号はRS−232C規格で　入力信号である。 なお、UARTの$\overline{CTS}$信号は$\overline{CTS}$=Lでデータ送信可能。 $\overline{CTS}$=Hでデータ送信不可(但し　コマンド　TxEN=1とする)

(注1)

3－3　バス信号

ＳＭ－Ｂ－８０Ｄに入出力するバス信号線の一般仕様を次に示す。

一般仕様

(1) 信号の論理

データ　　負論理

アドレス　　負論理

制御線　　負論理　　但し、デージー・チェーン制御線（ＩＥＩ、ＩＥＯ）は正論理とする。

(2) 信号のレベル

入力信号　　ＴＴＬコンパティブル

出力信号　　ＴＴＬコンパティブル

バス信号線の機能説明を表 3.2 に、また、バス信号一覧表を表 3.3 に示す。

表3. 2　　バス信号機能説明

（その１）

信 号 名	端子番号	機 能 説 明	信号方向
＋5V	1. 2. 3. 51.52.53	電源＋５Ｖ	入力
＋12V	5. 55	電源＋１２Ｖ	入力
＊CK/TG_1	6	ClocK/TriGger 1　この信号の反転信号が、Ｚ－８０ＣＴＣのＣＬＫ／TRG_1に印加される。信号として、ＣＴＣチャンネル１の外部クロックか、タイマ・トリガ信号を印加する。	入力 Ｌレベル アクティブ
＊ZC/TO_1	7	Zero Count/Time Out 1　ＣＴＣのＺＣ／TO_1出力の反転信号である。	出力 Ｌレベル アクティブ
＊CK/TG_2	8	ClocK/TriGger 2　この信号の反転信号がＣＴＣのＣＬＫ／TRG_2に印加される。信号として、ＣＴＣチャンネル２の外部クロックか、タイマ・トリガ信号を印加する。	入力 Ｌレベル アクティブ
＊ZC/TO_2	9	Zero Count/Time Out 2　ＣＴＣのＺＣ／TO_1出力の反転信号である。	出力 Ｌレベル アクティブ
＊CK/TG_3	10	ClocK/TriGger 3　この信号の反転信号がＣＴＣのＣＬＫ／TRG_3に印加される。信号としてＣＴＣチャンネル３の外部クロックか、タイマ・トリガ信号を印加する。	入力 Ｌレベル アクティブ
＊MRESET	11	Master RESET　スイッチS_1によりSM-B-80 Dをマニュアル・リセットすると、本信号は"L"レベルとなって出力するので、本信号を用いて外部装置を同時にリセットできる。また、本信号を外部から"L"レベルにすることによりSM-B-80 D内部をリセットできる。この場合、"L"レベルの期間SM-B-80 D内部はリセット状態にある。オープン・コレクタ出力。	入出力 Ｌレベル アクティブ
＊DEBUG	12	DEBUG　通常入力として使用される。本信号が"L"レベルの期間、アドレスの上位３ビット（AB_{15}、AB_{14}、AB_{13}）は強制的に１になる。したがって、本信号と＊MRESETに同時に"L"→"H"の信号を与えることにより、スイッチS_2に関係なくモニタをＥ０００Ｈからリスタートできる。S_2をオンにすると＊MRESETと同一極性の信号が出力する（この場合、オープン・コレクタ出力）。	入（出）力 Ｌレベル アクティブ
∅	62	Clock∅　ＣＰＵに供給するクロック信号と同一極性の出力信号。	出力

（その２）

信号名	端子番号	機能説明	信号方向
EXCLK	13	EXternal CLocK　外部クロックにより CPUを動作させる場合、この端子からクロック信号を供給する。この端子に加えられた信号の反転信号がCPUに印加される。なお内部クロックと、外部クロックの切換えはチェック・ピン端子∅CLKの配線により行う。	入力
XSCLK	63	eXternal Serial CLocK　外部シリアル・クロックにより UART 8251を動作させる場合、この端子からシリアル・クロック信号を供給する。通常8251のシリアル・クロックは CTCのZC/TO_0 出力を用いているので、外部シリアル・クロックを使用する場合、チェック・ピン端子S.CLKの配線を変更する必要がある。	入力
REC. DATA	14	RECeived DATA at terminal　コネクタJ_2のREC DATA端子信号に同じ。SM−B−80 Dから送出されるシリアル・データ信号。	出力 Hレベル アクティブ
TRANS. DATA	64	TRANSmitted DATA from terminal　コネクタJ_2のTRANS DATA 端子信号に同じ。SM−B−80 Dへ送出されてくるシリアル・データ信号。	入力 Hレベル アクティブ
＊DDIS	15	Driver DISable　この端子を"L"レベルにすることにより、データ・バッファのドライバ出力はトライステートになる。通常この端子データ・バッファの内部ドライバ制御信号が出力している。オープン・コレクタ出力。	入(出)力 Lレベル アクティブ
＊RENB	65	Receiver ENaBle　この端子を"L"レベルにすることにより、データ バッファのレシーバはイネーブルになる。通常、この端子は"H"レベルにプルアップされている。	入力 Lレベル アクティブ
IEI	20	Interrupt Enable In　本信号はプルアップされており、バッファ後 CTCのIEI入力に印加されている。デージー・チェーン接続で、CTCよりも割り込み優先レベルの高い装置がある場合、その装置のIEOを本端子に接続する。CTCを最高優先レベルに置く場合、本端子は開放（Hレベル）とする。	入力 Hレベル アクティブ
IEO	70	Interrupt Enable Out　SM−B−80D内のPIOよりも割り込み優先レベルの低い装置がある場合、その装置のIEI端子に本信号を印加する。PIO以上の装置に割り込みがある場合、本信号は"L"レベルとなる。	出力 Hレベル アクティブ

（その3）

信号名	端子番号	機能説明	信号方向
アドレス		Address CPUにバス・リクエストがかかっていない通常の使用状態（*BUSAK="H"）では、本信号はCPUのアドレス出力の反転信号に等しい。CPUにバス・リクエストが入力し、*BUSAK="L"となると、本信号はトライ・ステートとなる。 アドレス・バッファは双方向性であり、そのドライバは*BUSAK="H"でイネーブル、*BUSAK="L"でトライ・ステートになっている。レシーバにおいては、$*A_0$〜A_{11}は常にイネーブル、$*A_{12}$〜A_{15}は*BUSAK="L"のときのみイネーブルになる。このようにすることにより外部からSM−B−80D内のメモリをアクセスすることができる。	入出力 Lレベル アクティブ
	右参照	記号 $*A_0$ $*A_1$ $*A_2$ $*A_3$ $*A_4$ $*A_5$ $*A_6$ $*A_7$ $*A_8$ $*A_9$ $*A_{10}$ $*A_{11}$ $*A_{12}$ $*A_{13}$ $*A_{14}$ $*A_{15}$ ピン 22 72 23 73 24 74 25 75 26 76 27 77 28 78 29 79	
*WAIT	30	WAIT　本信号はプルアップされており、バッファ後CPUのWAIT端子に印加されている。メモリやI/O装置が低速であり、通常のクロック・サイクルでは応答できない場合に使用する。本信号を"L"レベルにすることにより、メモリやI/Oのアクセス時にウエイト状態が挿入される。	入力 Lレベル アクティブ
*BUSRQ	80	BUS ReQuest　本信号はプルアップされており、バッファ後、CPUのBUSRQ端子に印加されている。本信号はSM−B−80Dのアドレス出力、制御信号出力（*MREQ *IORQ $*M_1$ *RD *WR）をトライ・ステートにする場合に使用する。本信号を"L"レベルにすると、現在実行中の命令のマシン・サイクル（M_1又はM_2又はM_3）の完了後に*BUSAKが"L"レベルになり、アドレス出力、制御信号出力はトライ・ステートになる。	入力 Lレベル アクティブ
*NMIU	31	NMI for User　ユーザ用のNMI信号であり、本信号を"L"レベルにすると、現在実行中の命令完了後に割り込み、イネーブルF/Fの状態に関係なくCPUはNMIを受け付ける。NMI受け付け後　ダミーサイクル、PCの内容の退避サイクルを経てCPUは　0066Hから実行を始める。	入力 Lレベル アクティブ

（その4）

信号名	端子番号	機能説明	信号方向
＊INTU	81	INT for User　　ボード外部からのINT信号である。通常 $^{I}/_{O}$ からの割り込み信号として使用する。CPUの割り込みイネーブル $^{F}/_{F}$ がセットされている場合に限り本信号を"L"レベルにすると現在実行中の命令完了後にCPUは割り込み動作に入る。	入力 Lレベル アクティブ
＊BUSAK	34	BUS Aknowledge　　＊BUSRQが"L"レベルになるとCPUは現在実行中の命令のマシン・サイクル（M_1 又は M_2 又は M_3 ）の完了後に＊BUSAKを"L"レベルとする。＊BUSAKが"L"レベルになると制御出力（＊M_1　＊RFSH　＊RD　＊WR　＊MREQ　＊IORQ）、及びアドレス出力（＊A_0～A_{15}）はトライ・ステートになる。	出力 Lレベル アクティブ
＊HALT	84	HALT　　HALT命令を実行すると本信号は"L"レベルになり、同時にHALTランプが点灯する。	出力 Lレベル アクティブ
＊M_1	35	M_1　　本信号はCPUの$\overline{M_1}$出力をバッファしたものである。M_1 サイクルの命令コードのフェッチの期間　本信号は"L"レベルになる。本信号のバス・インターフェイスは双方向性バッファを用いており、ドライバは＊BUSAK＝"H"でイネーブル、＊BUSAK＝"L"でトライ・ステートとなる。また、レシーバは常にイネーブルになっており、＊M_1 はレシーバ経由でSM−B−80D内の各部に供給される。	出力 Lレベル アクティブ
＊RFSH	85	ReFreSH　　本信号はCPUのRFSH出力をバッファしたものである。ダイナミック・メモリに対するリフレッシュ・アドレスがCPUからアドレス・バスへ出力している期間、本信号は"L"レベルになる。本信号のバス・インターフェースは双方向性バッファを用いており、動作は＊M_1 と同じようになっている。	出力 Lレベル アクティブ
＊RD	36	ReaD　　本信号はCPUの$\overline{RD}$出力をバッファしたものである。CPUがメモリ又は $^{I}/_{O}$ 装置からデータを読み出す場合に、本信号は"L"レベルになる。本信号のバス・インターフェースは双方向性バッファを用いており、動作は＊M_1 と同じようになっている。	出力 Lレベル アクティブ
＊WR	86	WRite　　本信号はCPUの$\overline{WR}$出力をバッファしたものである。CPUがメモリ又は $^{I}/_{O}$ 装置へデータを書き込む場合に本信号は"L"レベルになる。本信号のバス・インターフェースは双方向性バッファを用いており、動作は＊M_1 と同じようになっている。	出力 Lレベル アクティブ

（その５）

信 号 名	端子番号	機　　能　　説　　明	信号方向
＊MREQ	37	Memory REQuest　本信号はＣＰＵのMREQ出力をバッファしたものである。本信号が“L”レベルになることによって、メモリの読み出しや書き込み動作でアドレス・バス上に正しいアドレス情報が出力していることがわかる。本信号のバス・インターフェースは双方向性バッファを用いており、動作は＊M_1 と同じようになっている。	出力 Ｌレベル アクティブ
＊ＩＯＲＱ	87	Input／Output ReQuest　本信号はＣＰＵのＩＯＲＱ出力をバッファしたものである。本信号が“L”レベルになるのは次の場合である。 ① Ｉ／Ｏ装置の読み出しや書き込み動作で、アドレス・バスの下位８ビット上に正しいアドレス情報が出力している期間。 ② ＣＰＵが割り込みを受け付け、割り込み応答ベクトルをデータ・バス上に乗せてもよいことを示している期間。 本信号のバス・インターフェースは双方向性バッファを用いており、動作は＊M_1 と同じようになっている。	出力 Ｌレベル アクティブ
データ	右 参 照	Data　本信号は双方向性のバッファ（反転）を経由してＣＰＵのデータ端子に接続している。双方向性のバッファのうち、レシーバは＊ＲＥＮＢによって制御され、＊ＲＥＮＢ＝“L”でイネーブル、＊ＲＥＮＢ＝“H”、または開放でトライステートとなる。一方、ドライバは、＊ＤＤＩＳ又は内部ドライバ制御信号によって制御され、いずれかが“L”レベルになるとトライ　ステートになる。ドライバがイネーブルになるのは、＊ＤＤＩＳ＝“H”かつ内部ドライバ制御信号＝“H”のときであり、内部ドライバ制御信号＝“H”となる条件は次のいずれかの場合である。 ① ＊MREQ又は＊ＩＯＲＱが“L”レベルであり、かつ＊M_1 ＊ＢＵＳＡＫ ＊ＲＤがいずれも“H”レベルのとき。 ② ＊ＲＦＳＨ＝“H”かつ＊ＲＤ＝“L”かつ＊ＭＲＥＱ＝“L”であり、ＳＢＣ上のいずれかのメモリがアクセスされたとき。 記号　＊D_0　＊D_1　＊D_2　＊D_3　＊D_4　＊D_5　＊D_6　＊D_7 ピン　42　92　43　93　44　94　45　95	入出力 Ｌレベル アクティブ
－１２Ｖ	46.　96	電源　－１２Ｖ	入力
ＧＮＤ	48. 49. 50 98.99. 100	接 地 線	

表3. 3　　バス信号一覧表

端子番号	信号名（部品名）	端子番号	信号名（配線名）	端子番号	信号名（部品名）	端子番号	信号名（配線面）
1	＋5V	51	＋5V	26	＊A_8	76	＊A_9
2	＋5V	52	＋5V	27	＊A_{10}	77	＊A_{11}
3	＋5V	53	＋5V	28	＊A_{12}	78	＊A_{13}
4		54		29	＊A_{14}	79	＊A_{15}
5	＋12V	55	＋12V	30	＊WAIT	80	＊BUSRQ
6	＊CK/TG_1	56		31	＊NMIU	81	＊INTU
7	＊ZC/TO_1	57		32		82	
8	＊CK/TG_2	58		33		83	
9	＊ZC/TO_2	59		34	＊BUSAK	84	＊HALT
10	＊CK/TG_3	60		35	＊M_1	85	＊RFSH
11	＊MRESET	61		36	＊RD	86	＊WR
12	＊DEBUG	62	∅	37	＊MREQ	87	＊IORQ
13	EXCLK	63	XSCLK	38		88	
14	REC DATA	64	TRANS DATA	39		89	
15	＊DDIS	65	＊RENB	40		90	
16		66		41		91	
17		67		42	＊D_0	92	＊D_1
18		68		43	＊D_2	93	＊D_3
19		69		44	＊D_4	94	＊D_5
20	IEI	70	IEO	45	＊D_6	95	＊D_7
21		71		46	−12V	96	−12V
22	＊A_0	72	＊A_1	47		97	
23	＊A_2	73	＊A_3	48	GND	98	GND
24	＊A_4	74	＊A_5	49	GND	99	GND
25	＊A_6	75	＊A_7	50	GND	100	GND

（注）

＊記号は"L" レベルで有効（active）になる意味であり、信号名の上に ―― （bar）記号をつけたものに同じ。

３－４　動作モードの選択

ＳＭ－Ｂ－８０Ｄではプログラム開発の用途だけでなく、他の種々の用途にも対処できるように各種の動作モードを選択できる機能を有している。これらの選択は大別するとラッピング端子の配線による方法とプラットホーム端子の配線による方法に分類できる。

３－４－１　ラッピング端子による方法

ボード上には３ケ所のラッピング個所があり、通常ラッピング配線によって動作モードの選択を行う。この方法により選択する動作モードとしては、システム・クロックの内部、または外部供給の切換え、シリアル・クロックの内部、または外部供給の切換え、ＴＴＹとＲＳ－２３２Ｃインターフェースの切り換えが行われる。これらの動作モードの選択法を表3.4に示す。

３－４－２　プラットホーム端子による方法

プラットホーム端子は１６ピンＩＣソケットに装着できる汎用のディスクリート部品配線台である。ＳＭ－Ｂ－８０ＤではこのプラットホームをＳ個使用しており、K_1 、K_2 、K_3 と名前をつけている。プラットホーム端子の外観を図3.20に示す。

K_1 によってボー・レートの選択、拡張 I/O ドライバ・ルーチンのモニタ管理の有無、及びシリアル I/O インターフェースからの割り込み制御の有無を指示できる。これらの設定法を表3.5、表3.6、表3.7に示す。

K_2 によってユーザＲＡＭ、ＯＳ　ＲＯＭのベース・アドレスを選択できるが、この設定法について表3.8にまとめる。

K_3 によってユーザＲＡＭの４Ｋバイト、１６Ｋバイトの選択、及び　ＯＳ　ＲＯＭのＥＰＲＯＭ、ＰＲＯＭの選択ができる。表3.9、表3.10にこれらの切り換え法を示す。

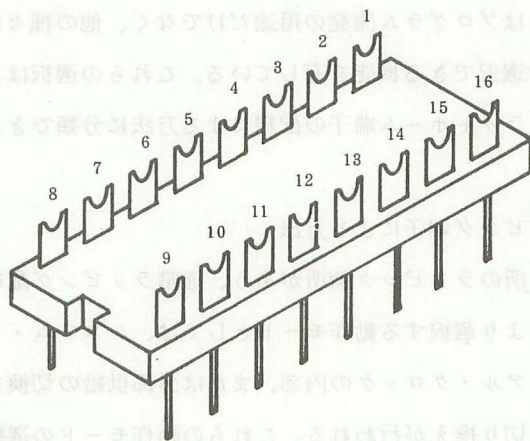



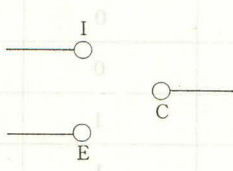
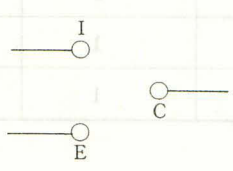
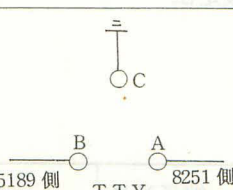
図3．20　プラット・ホーム端子
（ジャンパ端子）

項　　目	ボード上記号	結　　線　　法
システムクロック	∅. CLK	∅. CLK I / C / E 内部クロック使用時　I－C　結線 外部クロック使用時　E－C　結線 （出荷時I－C　結線）
シリアルクロック	S. CLK	S. CLK I / C / E 内部シリアル　クロック使用時　I－C　結線 外部シリアル　クロック　E－C　結線 （出荷時I－C　結線）
CTS接地	TTY	C / B / A 75189 側　TTY　8251 側 TTY使用時　A－C　結線 その他の場合　A－B　結線 （出荷時A－C　結線） （注）　B－Cを結線してはならない。

表 3. 4　　ラッピング端子結線法

ソケット	ボー・レート	K_1 端子番号（ピン）		
		6－11	5－12	4－13
K_1	110	0	0	0
	150	0	0	1
	300	0	1	0
	600	0	1	1
	1200	1	0	0
	2400	1	0	1
	4800	1	1	0
	9600	1	1	1

0：端子開放

1：端子接続

表 3. 5　ボー・レート設定法

ソケット	EPROM ＃3、＃4	K_1 端子番号	説明
K_1	汎用	7－10 開放	EPROM＃3、＃4を汎用の用途として使用できる。（出荷時このモード）
	拡張 I／O ドライバ・ルーチン用	7－10 接続	EPROM＃3、＃4を拡張 I／O ドライバ・ルーチン用として用い、そのエントリ・アドレス、ニモニックをモニタで管理する。

表 3. 6　拡張 I／O ドライバ・ルーチンの使用法

ソケット	割り込みの種類	K_1 結線法	
		割り込みあり	割り込みなし
K_1	Transmitter Ready (TR)	1－16 接続	13－16 接続
	Receiver Ready (RR)	2－15 接続	13－15 接続
	Transmitter Empty (TE)	3－14 接続	13－14 接続

（注） 本ボードの通常の使用法では割り込みなしの状態に結線する。
出荷時割り込みなしの状態に結線されている。

表 3. 7　シリアル・インターフェイスの割り込み

ソケット	4Kバイト RAM		16Kバイト RAM		4Kバイト ROM	
	ベース・アドレス	K_2 結線法	ベース・アドレス	K_2 結線法	ベース・アドレス	K_2 結線法
K_2	0000	5－15．1－16	0000	5－15－16	0000	5－9．1－10
	1000	5－15．2－16			1000	5－9．2－10
	2000	5－15．3－16			2000	5－9．3－10
	3000	5－15．4－16			3000	5－9．4－10
	4000	6－15．1－16	4000	6－15－16	4000	6－9．1－10
	5000	6－15．2－16			5000	6－9．2－10
	6000	6－15．3－16			6000	6－9．3－10
	7000	6－15．4－16			7000	6－9．4－10
	8000	7－15．1－16	8000	7－15－16	8000	7－9．1－10
	9000	7－15．2－16			9000	7－9．2－10
	A000	7－15．3－16			A000	7－9．3－10
	B000	7－15．4－16			B000	7－9．4－10
	C000	8－15．1－16	C000	8－15－16	C000	8－9．1－10
	D000	8－15．2－16			D000	8－9．2－10
	E000	8－15．3－16			E000	8－9．3－10
	F000	8－15．4－16			F000	8－9．4－10

（注） モニタ使用時4KバイトROMのベース・アドレスは、E000とする。
RAMのベース・アドレスは、通常0000とする。
出荷時　ROMはE000、RAMは0000になっている。

表3. 8　RAM／ROMのベース・アドレス設定法

ソケット	ダイナミックRAM	K_1 結線法			
K_3	4K RAM（4027 相当）	6−11	5−9	7−8−3	
	16K RAM（4116 相当）	5−12	6−11	7−9	8−10

表3. 9　4KRAM、16KRAMの切換え法

ソケット	EPROM／PROM	K_1 結線法		
K_3	EPROM（2708 相当）	1−16	2−15	3−14
		（出荷時この結線になっている）		
	PROM（注）	3−16	4−15	4−14

（注）バイポーラ PROMで2708とピン互換性あり　例 LH−7055

表3.10　EPROM、PROMの切換え法

4. ソフトウェア

4. 及び 5. はSM−B−80Dのソフトウェア　特にモニタの動作とその使用法について述べる。なお、ここで説明するモニタの機能は、2Kバイト　モニタ（V1.1）に関するものである。（2Kバイト　モニタの製品名は　LH−8S03P又はLH−8S03Eである。）

4−1　構　成

4−1−1　メモリ

SM−B−80D全体のメモリ・マップについては既に2.2において説明した（図2.3参照）ので、ここでは特にモニタに関連するメモリ領域のマップについて詳しく述べる。

モニタが関係するメモリ領域としては、E000〜EFFFのアドレス範囲のOS、ROM、及び　FF00〜FFFFのOS　RAM領域が挙げられる。

OS　ROM領域には2Kバイトのモニタ・プログラムが入っている。（オプション）

OS　RAM領域は256バイトの容量であり、モニタではこの領域を次の用途に使用している。OS　RAMのメモリ・マップを図4.1に示す。

(1) 一時データ退避用

(2) モニタ用スタック

(3) ユーザ用スタック

(4) ユーザ定義ニーモニックの記憶

(5) ユーザCPUレジスタ内容の記憶

ユーザ・プログラムのCPUレジスタ内容はOS　RAMのFFE6〜FFFFの範囲に退避されるが、そのメモリ・マップを図4.2に示す。

4−1−2　I/Oポート

モニタでは I/O ポート・アドレスとしてD0〜DFの範囲を使用している。本ボードでは、00〜CFのポート・アドレスをユーザ側に開放しており、D0〜FFをシステム側で使用、またはリザーブしている。この様子を図4.3に示す。

D0〜DFにより指定されるポートとして既に述べたように、PIO、CTC、UART、及び　モニタ・コマンドを制御するためのハードウェア信号（例　NMID）等が挙げられる。ポート・アドレスD0〜DFで指定されるこれらのバイト・データの各ビットの意味について次に簡単にまとめる。

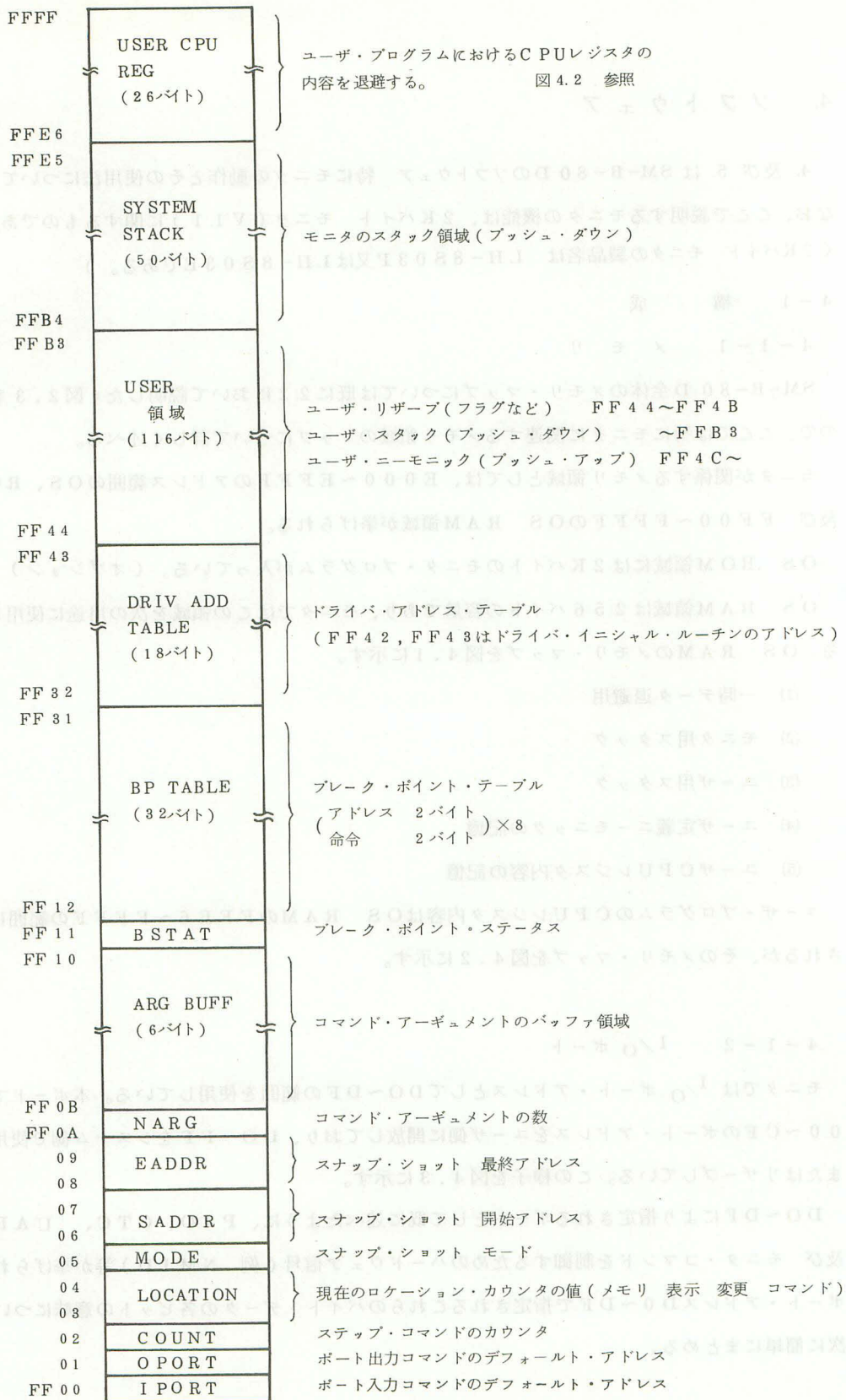



図4.1 OS RAM メモリ・マップ

アドレス	OS RAM	対応するニーモニック
FFFF	PCH	
FFFE	PCL	・PC
FFFD	A	・A
FFFC	F	・F
FFFB	I	・I
FFFA	IFF	・IF
FFF9	B	・B
FFF8	C	・C
FFF7	D	・D
FFF6	E	・E
FFF5	H	・H
FFF4	L	・L
FFF3	A′	・A′
FFF2	F′	・F′
FFF1	B′	・B′
FFF0	C′	・C′
FFEF	D′	・D′
FFEE	E′	・E′
FFED	H′	・H′
FFEC	L′	・L′
FFEB	IXH	
FFEA	IXL	・IX
FFE9	IYH	
FFE8	IYL	・IY
FFE7	SPH	
FFE6	SPL	・SP

（注）　ユーザRAMに割り当てられたアドレスは、対応するニーモニックにより直接に指定することができる。

・はドット(ASCII 2E)．′はASCII 27である。

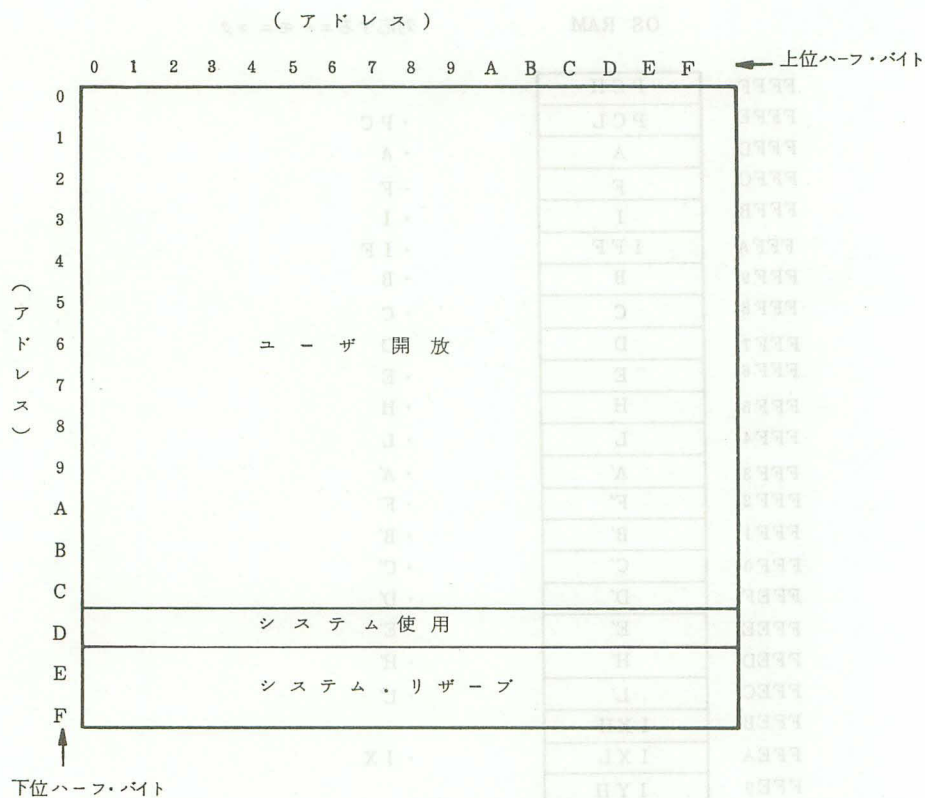
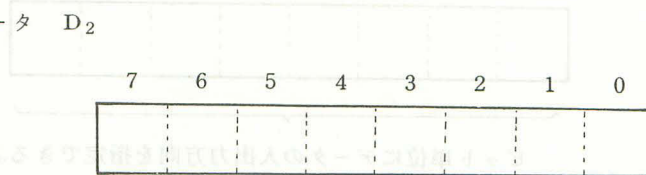
図4．2　ユーザCPUレジスタのメモリ・マップ

(注) ユーザ開放： ユーザ側で使用可能なI/Oポート．アドレス
　　 システム使用： モニタで使用しているI/Oポート・アドレス
　　 システム・リザーブ： 将来システムで使用予定のI/Oポート・アドレス

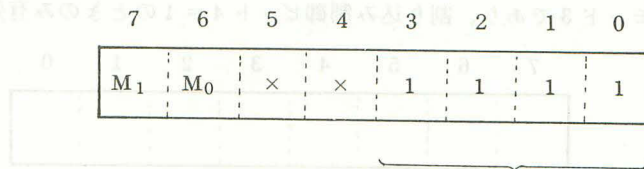
図4．3　I/Oポート・アドレス

〈ＰＩＯ〉

ポートＡ　データ　D_0

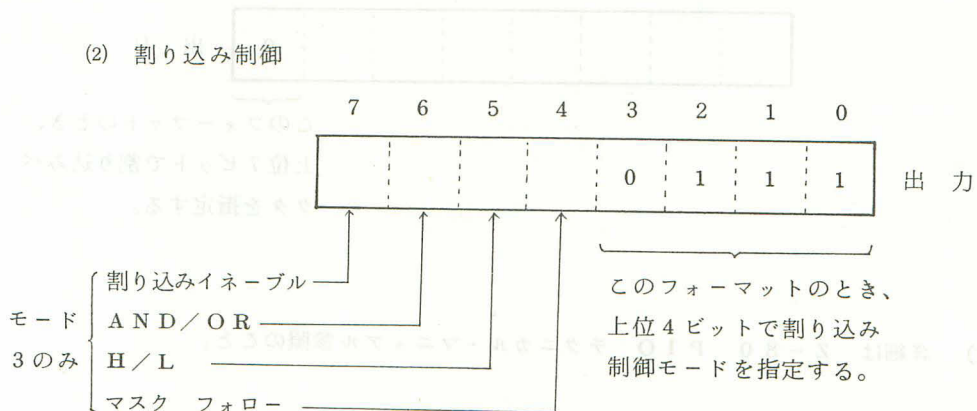
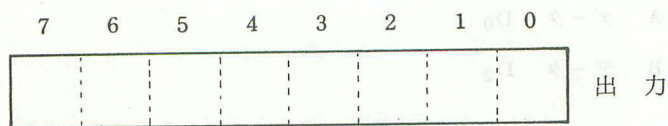
ポートＢ　データ　D_2

ポートＡ　コントロール　D_1

ポートＢ　コントロール　D_3

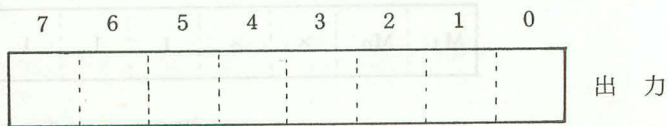
(1)　モード設定

このフォーマットのとき、M_1、M_0 でＰＩＯの動作モードを指定する。

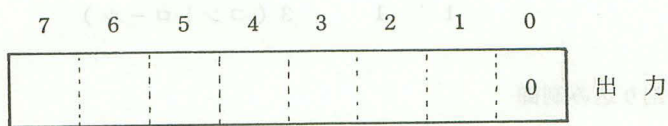
M_1	M_0	モード
0	0	0（出力）
0	1	1（入力）
1	0	2（双方向）
1	1	3（コントロール）

(2)　割り込み制御

このフォーマットのとき、上位４ビットで割り込み制御モードを指定する。

(3) データの方向（モード3のみ）

ビット単位にデータの入出力方向を指定できる。（1：入力、
0：出力）

この指定はモード3指定後、同一ポートへの始めての書き込み時に上記データを書き込むことにより行う。

(4) マスク（モード3であり、割り込み制御ビット4＝1のときのみ有効）

各ビットが割り込み発生の要因となるかどうかの指定を行う。
（1：割り込みに関係しない、0：割り込みに関係する）
この指定は割り込み制御ビット＝1のとき、次にこのポートへ書き込むデータによって行う。

(5) 割り込みベクタ

(注) 詳細は　Z－80　PIO　テクニカル・マニュアル参照のこと。

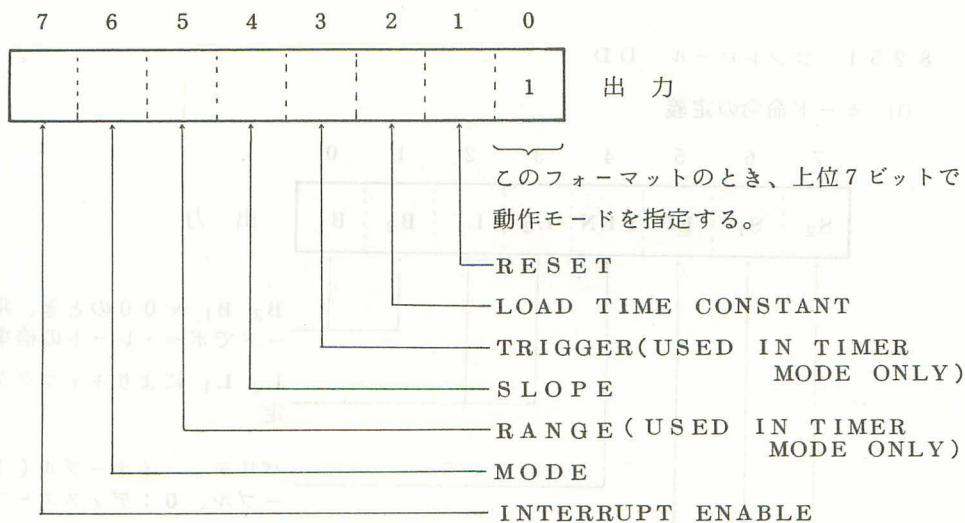
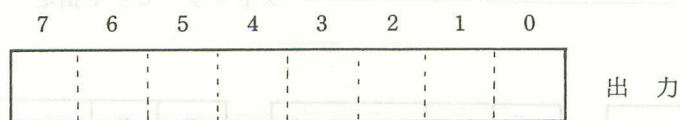
〈ＣＴＣ〉

チャンネル０　　Ｄ８

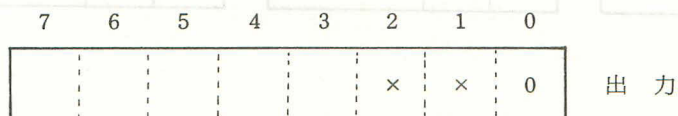
１　　Ｄ９

２　　ＤＡ

３　　ＤＢ

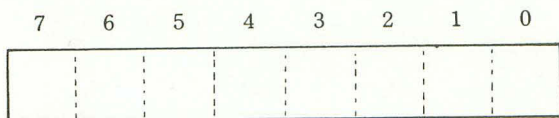
(1)　動作モードの指定

(2)　時定数の設定

上記第１項のＬＯＡＤ　ＴＩＭＥ　ＣＯＮＳＴＡＮＴ＝１のとき、続いて書き込む８ビット・データにより、時定数を設定できる。

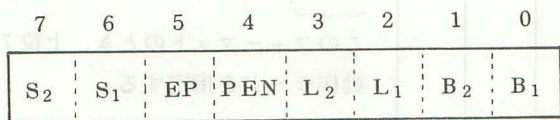
(3)　割り込みベクタ

7	6	5	4	3	2	1	0	
					×	×	0	出　力

このフォーマットのとき、上位５ビットで割り込みベクタを指定する。チャンネル０のベクタを指定すると、チャンネル1.2.3のベクタは２バイト間隔で自動的に決まる。

(注)　詳細はＺ－８０　ＣＴＣテクニカル・マニュアル参照のこと。

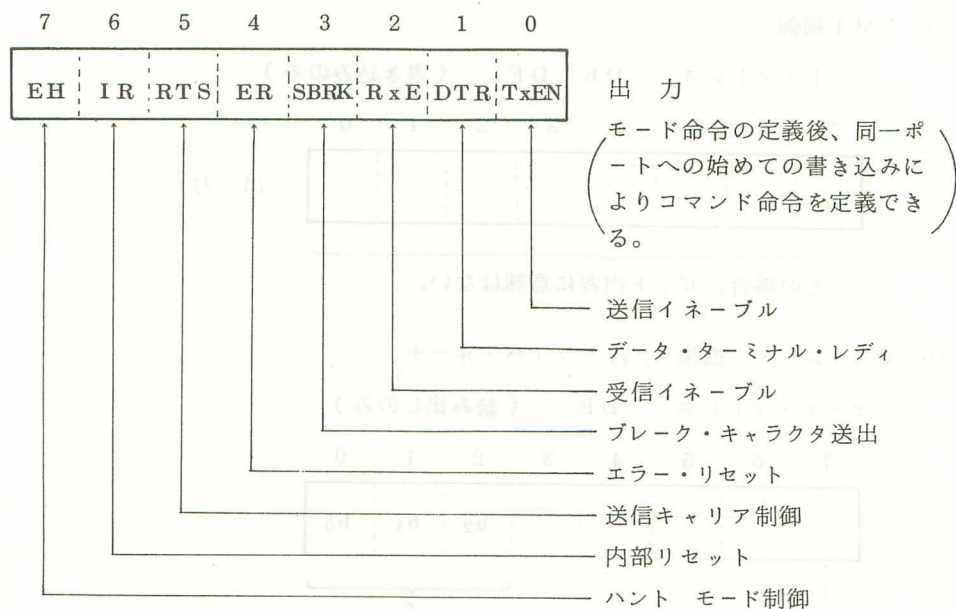
〈 UART 〉

8251 データ DC

8251 コントロール DD

(1) モード命令の定義

S_2	S_1	ストップ・ビット数
0	0	╳
0	1	1ビット
1	0	1 1/2 ビット
1	1	2ビット

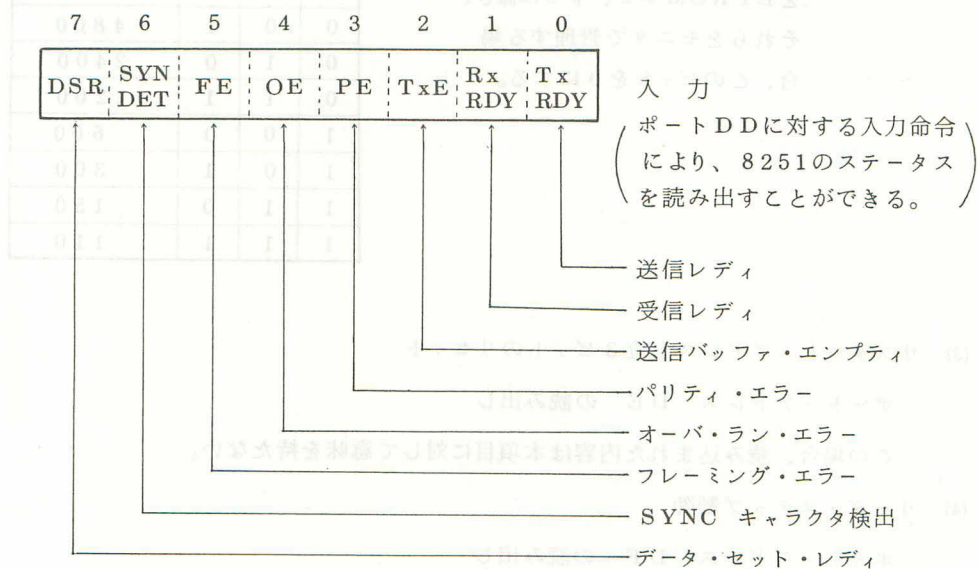
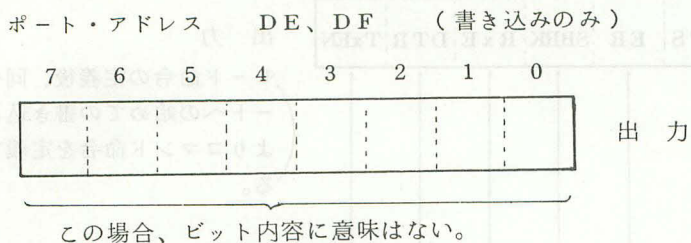
L_2	L_1	キャラクタ長
0	0	5ビット
0	1	6ビット
1	0	7ビット
1	1	8ビット

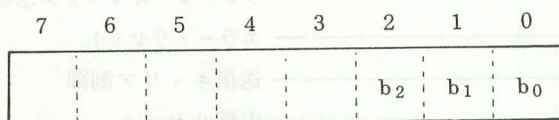
B_2	B_1	ボーレート
0	0	╳
0	1	1 ×
1	0	16 ×
1	1	64 ×

(2) コマンド命令の定義

(3) ステータスの読み出し

〈モニタ・コマンド制御〉

(1) NMI制御

ポート・アドレス　DE、DF　（書き込みのみ）

(2) ボー・レート、拡張 I/O ドライバ・ルーチン

ポート・アドレス　DE　（読み出しのみ）

拡張 I/O ドライバ・ルーチンをEPROM＃2、＃3に置き、それらをモニタで管理する場合、このビットを0にする。

b_2	b_1	b_0	ボー・レート
0	0	0	9600
0	0	1	4800
0	1	0	2400
0	1	1	1200
1	0	0	600
1	0	1	300
1	1	0	150
1	1	1	110

(3) リスタート・アドレス上位3ビットのリセット

ポート・アドレス　DE　の読み出し

この場合、読み込まれた内容は本項目に対して意味を持たない。

(4) リーダ・ステップ制御

ポート・アドレス　DF　の読み出し

リーダ・ステップ制御用のフリップ・フロップはポート・アドレス DFに対する読み出しによってセットされる。この場合、読み込まれた内容は本項目に対して意味を持たない。

4－1－3　ゼネラル・フロー

2Kバイト・モニタのゼネラル・フローを図4.4に示す。

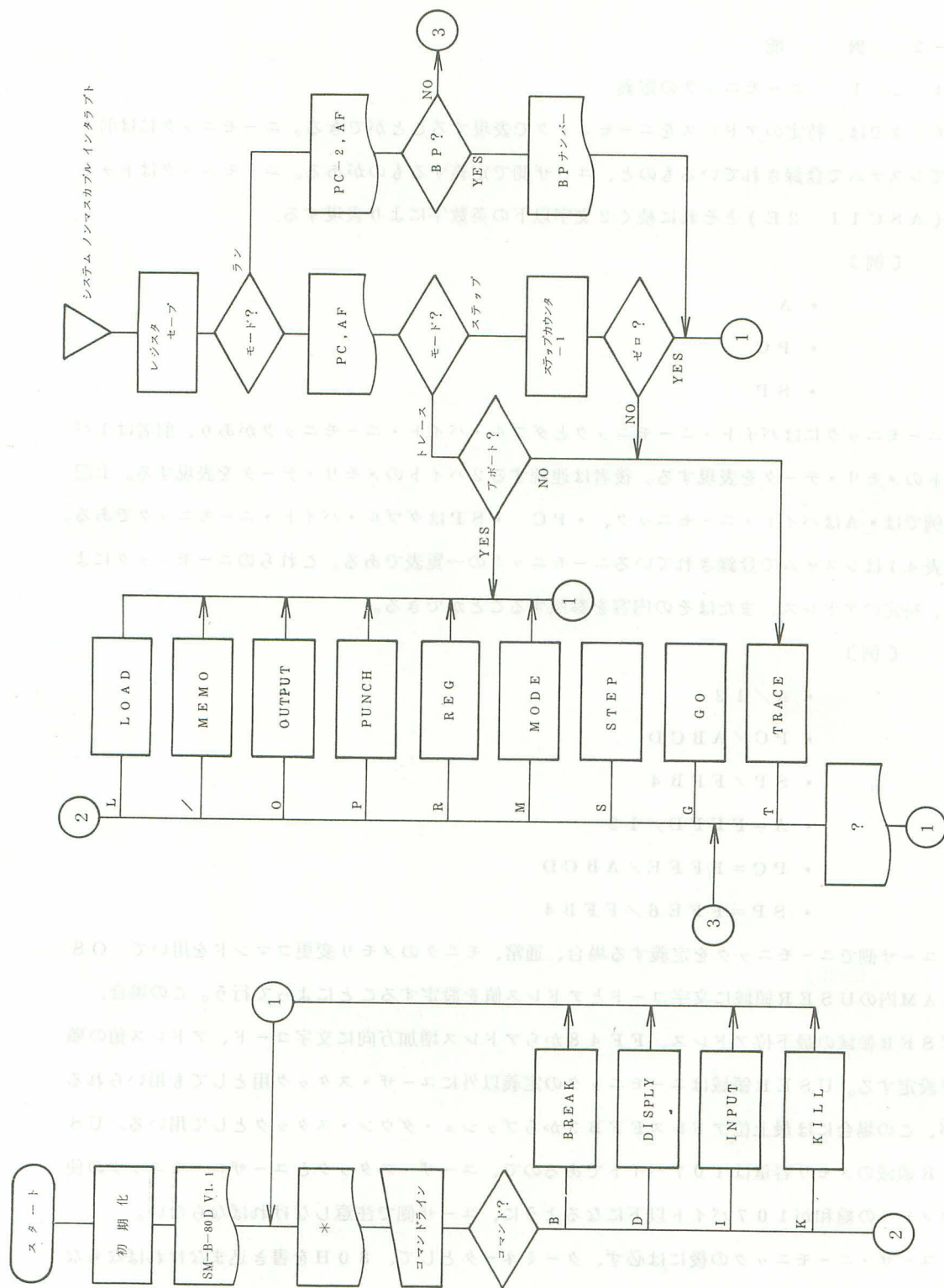



図4.4　2K　モニタ　フローチャート

4－2　　機　　　能

4－2－1　　ニーモニックの定義

モニタでは、特定のアドレスをニーモニックで表現することができる。ニーモニックには前もってシステムで登録されているものと、ユーザ側で定義するものがある。ニーモニックはドット・（ＡＳＣＩＩ　２Ｅ）とそれに続く２文字以下の英数字により表現する。

〔例〕

- Ａ
- ＰＣ
- ＳＰ

ニーモニックにはバイト・ニーモニックとダブル・バイト・ニーモニックがあり、前者は１バイトのメモリ・データを表現する。後者は連続する２バイトのメモリ・データを表現する。上記の例では・Ａはバイト・ニーモニック、・ＰＣ　・ＳＰはダブル・バイト・ニーモニックである。

表 4.1はシステムで登録されているニーモニックの一覧表である。これらのニーモニックにより、特定のアドレス、またはその内容を参照することができる。

〔例〕

- Ａ／１２
- ＰＣ／ＡＢＣＤ
- ＳＰ／ＦＦＢ４
- Ａ＝ＦＦＦＤ／１２
- ＰＣ＝ＦＦＦＥ／ＡＢＣＤ
- ＳＰ＝ＦＦＥ６／ＦＦＢ４

ユーザ側でニーモニックを定義する場合、通常、モニタのメモリ変更コマンドを用いて　ＯＳ　ＲＡＭ内のＵＳＥＲ領域に文字コードとアドレス値を設定することによって行う。この場合、ＵＳＥＲ領域の最下位アドレス、ＦＦ４８からアドレス増加方向に文字コード、アドレス値の順で設定する。ＵＳＥＲ領域はニーモニックの定義以外にユーザ・スタック用としても用いられるが、この場合には最上位アドレスＦＦＢ３からプッシュ・ダウン・スタックとして用いる。ＵＳＥＲ領域のメモリ容量は１０７バイトであるので、ユーザ・スタックとユーザニーモニックの使用メモリの総和が１０７バイト以下になるように、ユーザ側で注意しなければならない。

ユーザ・ニーモニックの後には必ず、ターミネータとして、８０Ｈを書き込まなければならない。図４．５にＵＳＥＲ領域の使用の様子を示す。

Ｉ／Ｏチャンネルとして拡張用Ｉ／Ｏドライバ・ルーチン使用する場合、ＲＯＭ　＃３、＃４

ニーモニック	アドレス	意　　　　味
・PC	FFFE	ユーザ・プログラム　CPU　レジスタ　PC
・A	FFFD	〃　〃　〃　〃　A
・F	FFFC	〃　〃　〃　〃　F
・I	FFFB	〃　〃　〃　〃　I
・IF	FFFA	〃　〃　〃　〃　IF
・B	FFF9	〃　〃　〃　〃　B
・C	FFF8	〃　〃　〃　〃　C
・D	FFF7	〃　〃　〃　〃　D
・E	FFF6	〃　〃　〃　〃　E
・H	FFF5	〃　〃　〃　〃　H
・L	FFF4	〃　〃　〃　〃　L
・A′	FFF3	〃　〃　〃　〃　A′
・F′	FFF2	〃　〃　〃　〃　F′
・B′	FFF1	〃　〃　〃　〃　B′
・C′	FFF0	〃　〃　〃　〃　C′
・D′	FFEF	〃　〃　〃　〃　D′
・E′	FFEE	〃　〃　〃　〃　E′
・H′	FFED	〃　〃　〃　〃　H′
・L′	FFEC	〃　〃　〃　〃　L′
・IX	FFEA	〃　〃　〃　〃　IX
・IY	FFE8	〃　〃　〃　〃　IY
・SP	FFE6	〃　〃　〃　〃　SP
・CI	FF32	コンソール入力チャンネル，通常・TKに設定されている
・CO	FF34	コンソール出力チャンネル，通常・TTに設定されている
・OI	FF36	オブジェクト入力チャンネル，通常・TRに設定されている
・OO	FF38	オブジェクト出力チャンネル，通常・TTに設定されている
・SI	FF3A	ソース入力チャンネル，通常・TRに設定されている
・SO	FF3C	ソース出力チャンネル，通常・TTに設定されている
・TK	E744	TTYキー・ボード・ドライバのエントリ・アドレス
・TT	E757	TTYタイプ・ヘッド・ドライバのエントリ・アドレス
・TR	E742	TTYテープ・リーダ・ドライバのエントリ・アドレス
・AS	C000	アセンブラのエントリ・アドレス（LH−8S01E，PROM版）
・ED	D1D0	エディタのエントリ・アドレス（LH−8S02E，PROM版）

表4．1　システム登録ニーモニック一覧表

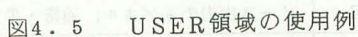
① ユーザ・プログラム内でスタックを使用する場合、ユーザ・プログラム内でSPをFFB4に初期設定しなければならない。

② ユーザ・ニーモニックとしてX，PA，25を定義し、そのアドレスを各々D100，D230，D345とする場合、メモリ変更コマンドによりUSER領域に左のように書き込む。

FFB3

ユーザ・スタック

アドレス	上位	下位	説明
FF58	8	0	ターミネータ
FF57	D	3	アドレス D345
FF56	4	5	
FF55	3	5	ASCII文字5
FF54	3	2	ASCII文字2
FF53	D	2	アドレス D230
FF52	3	0	
FF51	4	1	ASCII文字A
FF50	5	0	ASCII文字P
FF4F	D	1	アドレス D100
FF4E	0	0	
FF4D	2	0	ASCII文字 SP(空白)
FF4C	5	8	ASCII文字 X

図4．5　USER領域の使用例

にもしユーザ・ニーモニックが定義されているならば、その内容がＵＳＥＲ領域のアドレスＦＦ４８から増加方向にコピーされる。

４−２−２　I／O　チャンネル

モニタではI／O装置の制御にチャンネルという考え方を採用している。このモニタの機能（例えば、プログラムのロード、パンチ、デバッグ）を利用する場合、ソフトウェア上では、外部装置と各コマンド処理ルーチンはチャンネルを経由してインターフェースされる。次に示す6個のチャンネルが定義されており、モニタではこれ以外のチャンネルを使用することはできない。

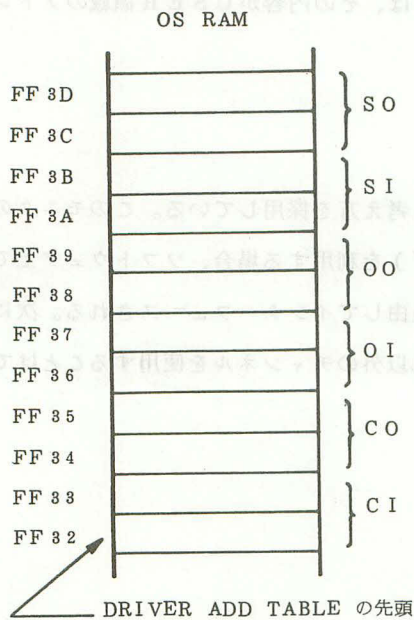
ＣＩ　コンソール入力チャンネル
ＣＯ　コンソール出力チャンネル
ＯＩ　オブジェクト入力チャンネル
ＯＯ　オブジェクト出力チャンネル
ＳＩ　ソース入力チャンネル
ＳＯ　ソース出力チャンネル

チャンネルには、ソフトウェア・ドライバ・ルーチン（のアドレス）を割り当てる必要があるが、この割り当てには次の３種類の方法のいずれかを選ぶ。

① モニタ内に前もって用意してあるドライバ・ルーチンを割り当てる。
② ＲＯＭ＃３、＃４に用意してあるドライバ・ルーチンを割り当てる。
③ 一般のＲＯＭに用意してあるドライバ・ルーチンを割り当てる。

①、②の方法では、ドライバ・ルーチンの割り当ては自動的に行うことができるが、③の方法では、メモリ内容を変更するコマンドにより、ＯＳ　ＲＡＭのＤＲＩＶＥＲ　ＡＤＤＲ　ＴＡＢＬＥを書き換えなければならない。

①の方法では、モニタ内のI／Oのドライバ・ルーチンを使用する。これらのドライバ・ルーチンとして、ＴＴＹ（ＲＳ２３２Ｃ相当の装置）のキー・ボード、タイプ・ヘッド、テープ・リーダの各ドライバ・ルーチンが用意されており、各々のルーチンの先頭アドレスは、・ＴＫ、・ＴＴ、・ＴＲと定義されている。通常、ＣＩには・ＴＫを、ＣＯ、ＯＯ、ＳＯには・ＴＴを、また　ＣＩ、ＳＩには・ＴＲを割り当てている。①の方法が選ばれるとモニタによりＤＲＩＶＥＲ　ＡＤＤＲ　ＴＡＢＬＥには自動的に・ＴＫ、・ＴＴ、・ＴＴがコピーされる。図４．６に ①の方法を示す。

②の方法はＴＴＹ（または　ＲＳ２３２Ｃ相当の装置）以外のI／O装置をチャンネルとする場合に用い、チャンネルへのドライバ・ルーチンのアドレス割り当ては自動的に行うことができ

OS RAM
FF 3D
FF 3C
FF 3B
FF 3A
FF 39
FF 38
FF 37
FF 36
FF 35
FF 34
FF 33
FF 32
SO
SI
OO
OI
CO
CI
DRIVER ADD TABLE の先頭
(注)
OS RAMにはモニタにより左のように
TTYのドライバ・ルーチンの先頭アド
レスがコピーされる。

OS ROM
#2
・TT(タイプ・ヘッド・ドライバ・ルーチンの先頭アドレス)
・TK(キー・ボード・ドライバ・ルーチンの先頭アドレス)
・TR(テープ・リーダ・ドライバ・ルーチンの先頭アドレス)

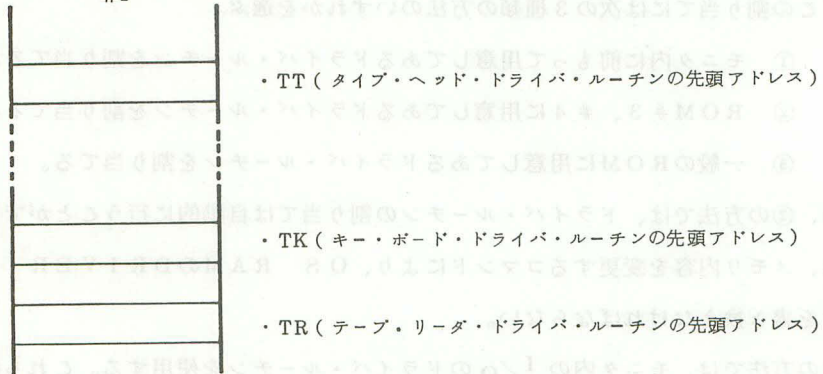



図4．6 ①の方法

る。但し、この場合には、ＯＳ　ＲＯＭ　＃３、＃４にチャンネルに対応するドライバ・ルーチンとその先頭アドレスを書き込み、かつ、ボード上のジャンパ端子K_1 の７－１０を接続しなければならない。表3.6参照。図４．７に②の方法を示す。なお、 この モニタでは、 ②の方法を拡張 I/O ドライバ・ルーチンによる方法と呼んでいる。

③の方法も②の方法と同じようにＴＴＹ（または　ＲＳ２３２Ｃ相当の装置）以外の I/O 装置をチャンネルとする場合に用いるが、チャンネルへのドライバ・ルーチンのアドレス割り当てはメモリ内容の変更コマンドを用いて行う。ユーザ側では I/O ドライバ・ルーチンをＲＯＭに用意しておき（この場合のＲＯＭは、ＯＳ　ＲＯＭ　＃３　＃４でもよい）、そのエントリ・アドレスをＯＳ　ＲＡＭのＤＲＩＶＥＲ　ＡＤＤ　ＴＡＢＬＥに書き込む。図4．8に③の方法を示す。

②、③の方法では I/O 装置のイニシャル・ルーチンが必要になる。②の方法では、イニシャル・ルーチンの先頭アドレスも自動的にＯＳ　ＲＡＭにコピーされる。図４．７、図４．８ではイニシャル・ルーチンの先頭アドレスを記号G_1 G_2 G_3 G_4 で与えているが、このアドレス以降にユーザ側でイニシャル・ルーチン（サブルーチン）のプログラムを用意しなければならない。このイニシャル・ルーチンはシステムがリセットされるたびにコールされる。また、ユーザが定義したドライバ・ルーチンのフラグ用としてＯＳ　ＲＡＭのＦＦ４４ ～ＦＦ４Ｂまでの８バイトをリザーブする。

OS RAM
8 0
ターミネータ
ユーザ
ニーモニック
FF43
42
G1 G2
G3 G4
イニシャル・ルーチンの先頭アドレス
3D
3C
3B
3A
39
38
37
36
35
34
33
FF32
F1 F2
F3 F4
E1 E2
E3 E4
D1 D2
D3 D4
C1 C2
C3 C4
B1 B2
B3 B4
A1 A2
A3 A4
SO
SI
OO
OI
CO
CI
DRIV ADD TABLEの先頭
（注１） ユーザ領域の先頭以後にユーザ・リザーブ（フラグなど、８バイト固定）、OS ROMのユーザ・ニーモニックおよびターミネータが自動的にコピーされる。
（注２） DRIV ADD TABLE の先頭以後にOS ROMに用意されている拡張I/Oドライバの先頭アドレスが、自動的にコピーされる。
OS ROM #3, #4
B1 B2 B3 B4
拡張I/Oドライバ・ルーチンの先頭
A1 A2 A3 A4
拡張I/Oドライバ・ルーチンの先頭
ユーザ・ニーモニック・ターミネータ
8 0
ユーザ・ニーモニック
（最大７個）
E812
11
10
拡張I/Oドライバのイニシャル・ルーチンのアドレス
0B
0A
09
08
07
06
05
04
03
02
01
E800
拡張I/Oドライバの先頭アドレス
（SO）
（SI）
（OO）
（OI）
（CO）
（CI）
ボード上のジャンパ端子
K1の７－10接続のこと

図4．7 ②の方法

ROM

B1 B2 B3 B4　　I/Oドライバ・ルーチンの先頭（CO）

C1 C2 C3 C4　　I/Oドライバ・ルーチンの先頭（OI）

A1 A2 A3 A4　　I/Oドライバ・ルーチンの先頭（CI）

D1 D2 D3 D4　　I/Oドライバ・ルーチンの先頭（OO）

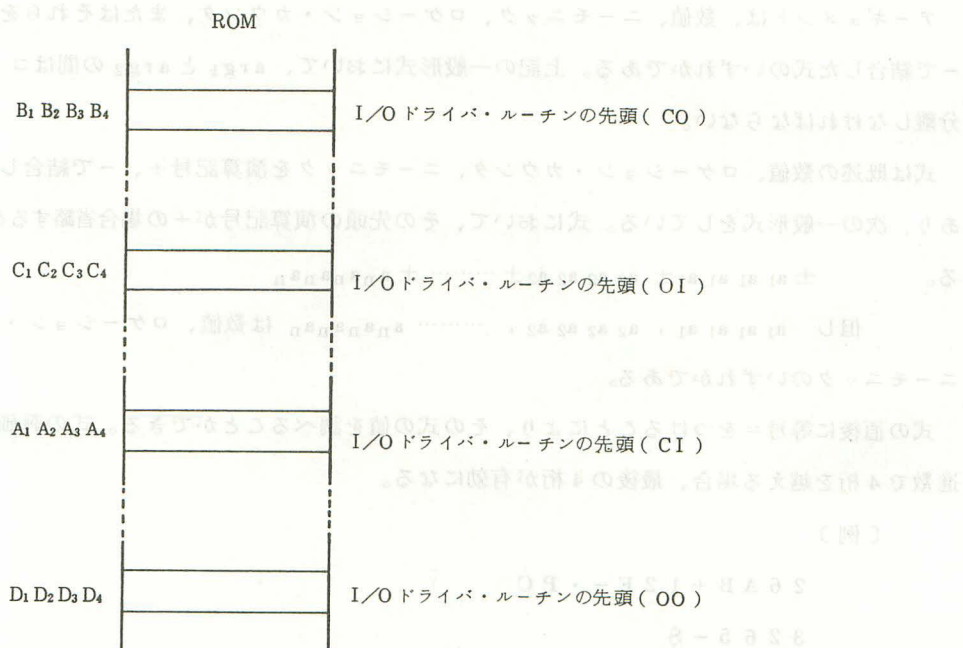
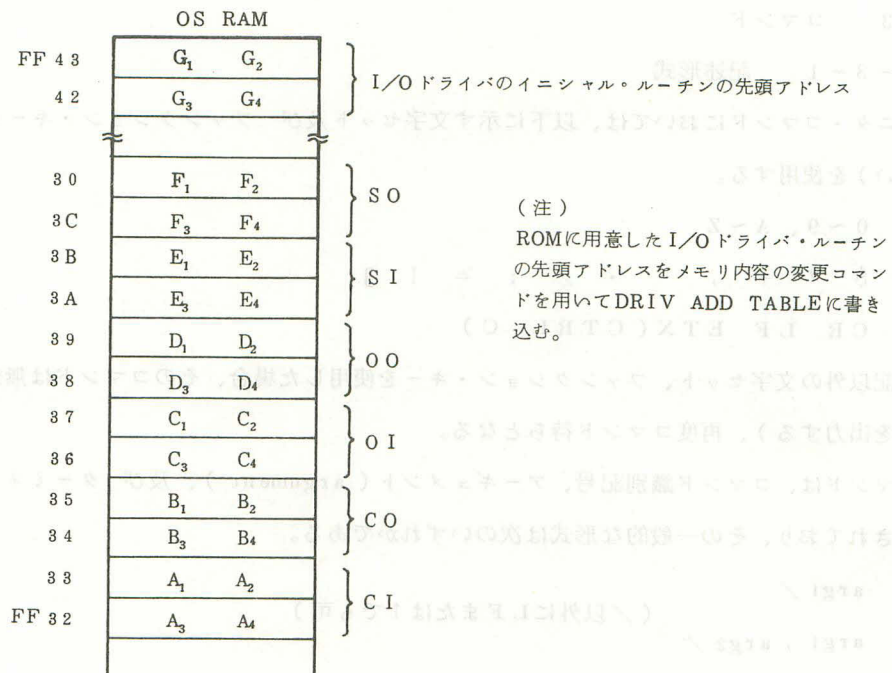



図4．8　③の方法

4－3　　コマンド

4－3－1　　記述形式

モニタ・コマンドにおいては、以下に示す文字セット及び　ファンクション・キー（印刷出力しない）を使用する。

０～９、Ａ～Ｚ

＄　▼　＋　，　－　・　／　；　＝　↑　］

ＣＲ　ＬＦ　ＥＴＸ（ＣＴＲＬ　Ｃ）

上記以外の文字セット、ファンクション・キーを使用した場合、そのコマンドは無効となり（？を出力する）、再度コマンド待ちとなる。

コマンドは、コマンド識別記号、アーギュメント（argument）、及び　ターミネータにより構成されており、その一般的な形式は次のいずれかである。

arg1 ／

arg1 ，arg2 ／　　　（／以外にＬＦまたは↑でも可）

arg1 ；Ｃ

arg1 ，arg2 ；Ｃ

但し、arg_1、arg_2 はアーギュメント、Ｃはコマンド記号

アーギュメントは、数値、ニーモニック、ロケーション・カウンタ、またはそれらを演算子＋、－で結合した式のいずれかである。上記の一般形式において、arg_1 と arg_2 の間はコンマ，で分離しなければならない。

式は既述の数値、ロケーション・カウンタ、ニーモニックを演算記号＋、－で結合したものであり、次の一般形式をしている。式において、その先頭の演算記号が＋の場合省略することができる。　　$\pm a_1 a_1 a_1 a_1 \pm a_2 a_2 a_2 a_2 \pm \cdots\cdots \pm a_n a_n a_n a_n$

但し　$a_1 a_1 a_1 a_1$，$a_2 a_2 a_2 a_2$，……… $a_n a_n a_n a_n$ は数値、ロケーション・カウンタ、ニーモニックのいずれかである。

式の直後に等号＝をつけることにより、その式の値を調べることができる。式の評価値が１６進数で４桁を越える場合、最後の４桁が有効になる。

〔例〕

２６ＡＢ＋１２Ｆ－・ＰＣ

３２６５－＄

３２６５－＄＝９１　　　　（＄の値が３２９４とする）

ＦＦＦＦ＋ＡＢＣＤ＝ＡＢＣＤ

4－3－2　アーギュメント

アーギュメントは４桁の１６進数を表現し、次のいずれかの形式をしている。

数　値

ロケーション・カウンタ（記号　＄）

ニーモニック

式

数値は最大４桁の１６進数である。先行する０は省略でき、５桁以上の１６進数を記述した場合、最後の４桁の数値が有効になる。使用できる文字は、０～９、Ａ～Ｆである。

〔例〕

２６ＡＢ

３Ｆ

００１０

１０

７３２６５　　　（３２６５とみなされる）

ロケーション・カウンタは、メモリ内容の変更、表示コマンドにおいてのみ意味を持つ。ロケーション・カウンタは、現在示しているメモリ・アドレスの次のアドレスを示し、ドル記号＄によって表示する。ロケーション・カウンタを用いると相対ジャンプのオフセットを現在のメモリ・アドレスを意識しないで求めることができる。

〔例〕

３２６５－＄　　（ロケーション・カウンタが３２９４であるとすると左の式の値は９１となる）

ニーモニックは既に述べたように、ドット記号とそれに続く２文字により表わす。ニーモニックにはシステムで既に登録しているものと（表4.1参照）、ユーザ側で定義するものがある。ニーモニックには、１バイト・ニーモニックと２バイト・ニーモニックがあり、各々２桁、４桁の１６進数を処理する。

〔例〕

・ＰＣ

・ＡＸ　　　（ユーザ定義ニーモニック）

4－3－3　　コマンド識別記号

コマンド識別記号（以下　コマンド記号という）は、セミコロン；とアルファベット１文字か

ら構成されており、アルファベット記号はコマンド機能に対応して前もって決められている。但し、メモリ内容の表示、変更コマンドでは、特定のコマンド記号を持たず、スラッシュ／で識別する。

コマンドの入力は、コマンド待ちの状態（行の先頭にモニタが＊を印刷出力した後）で行い、一般的には、アーギュメントに続けてコマンド記号をキー入力する。

〔例〕

＊１２３４；Ｂ

＊；Ｄ

＊１２３４，５６７８；Ｐ

＊１２３４／

表4．2にコマンド記号の一覧表を示す。

４－３－４　　ターミネータ

ターミネータはメモリ内容の表示、変更コマンド、及び　ユーザ・レジスタの表示変更コマンドにおいて有効である。ターミネータとして、carriage return（以下↓と記述） up arrow（以下↑と記述）、line feed（以下ＬＦと記述）があり、各々、次の機能を持っている。

① carriage return（↓）

現在のコマンドを完了した状態で↓を入力すると、次のコマンド待ちとなる。

〔例〕　＊１２３４／ＦＦ↓

＊　　　　　…………次のコマンド待ち

② up arrow（↑）

現在のコマンドを完了した状態で↑を入力すると、現在のメモリ・アドレスを－１して同一コマンドを実行し、次のターミネータ入力待ちとなる。

〔例〕　＊１２３４／ＦＦ↑

＊１２３３／００…………次のターミネータ待ち

③ line feed（ＬＦ）

現在のコマンドを完了した状態でＬＦを加入すると、現在のメモリ・アドレスを＋１して同一コマンドを実行し次のターミネータ入力待ちとなる。

〔例〕　＊１２３４／ＦＦ（ＬＦ）

＊１２３５／０Ｆ　………次のターミネータ待ち

表 4.2　コマンド一覧表

機　能	コ　マ　ン　ド	機　能　説　明
メモリ・レジスタの表示	arg_1/nn	arg_1 で指定されるメモリ・アドレス、またはユーザCPUの内容を2桁の16進数で/の直後に表示する。表示後ターミネータ待ち。
メモリ・レジスタの変更	arg_1/nn mm	arg_1 で指定されるメモリ・アドレス、またはユーザCPUレジスタの内容を変更する場合に使用する。変更は上記コマンドにより表示された数値に続いて、希望する数値（mm）を16進数で入力し、さらにターミネータを入力することによって行う。
ブレイク・ポイントの設定解除	arg_1 ; nB	arg_1 がある場合、ブレイク・ポイントの設定を行う。nは0〜7で0は省略可能。このコマンドによりアドレス arg_1 に識別番号nのブレイク・ポイントを設定する。arg_1 を省略すると、n番のブレイク・ポイントを解除する。
ブレイク・ポイントの表示	; D	現在設定されているブレイク・ポイントの識別番号と、そのアドレス（ブレイク・ポイント・アドレス）をnの順に表示する。
ユーザ・プログラムの実行	arg_1 ; G	arg_1 で指定されるメモリ・アドレスよりプログラム（ユーザ・プログラム）を実行する。arg_1 を省略した場合、現在のPC（ユーザ・CPUレジスタ）で示されるアドレスから実行する。
ポート入力	arg_1 ; nI	arg_1 ＋nで指定されるポートからデータ（1バイト）を読み込み表示する。nを省略した場合は、n＝0と等価である。nは0〜255の10進数とする。
ポート出力	arg_1 , arg_2 ; nO	arg_1 ＋nで指定されるポートへ arg_2 で示される1バイトデータを書き込む。nの意味は上記ポート入力の場合と同じ。
ブレイク・ポイントの全解除	; K	現在設定されているすべてのブレイク・ポイントを解除する。
ステップ	arg_1 ; nS	arg_1 で指定されるメモリ・アドレスよりnステップ実行させ、各ステップ毎にPC、AFの内容を印刷出力する。nを省略すると1ステップ動作。
トレース	arg_1 ; T	arg_1 で指定されるメモリ・アドレスよりトレースする。トレースはCTRLCのキー入力、またはブレイク・ポイント・アドレスにおいて終了し、コマンド待ちとなる。arg_1 を省略した場合、現在のPCの値よりトレースする。
プログラムのロード	; L	オブジェクト・チャンネルよりインテル標準16進フォーマットのオブジェクト・プログラムをメモリへロードする。
プログラムのパンチ	arg_1＋arg_2 ; P	arg_1、arg_2 で指定されるメモリ・アドレスの範囲の内容をオブジェクト・チャンネルに、インテル標準16進フォーマットで出力する。
レジスタの表示	; R	すべてのユーザCPUレジスタの内容を表示する。
メモリ・ブロックの表示	arg_1 , arg_2 /	arg_1、arg_2 で指定されるメモリ・アドレスの範囲の内容を2桁の16進数で表示する。
表示モードの指定	; nM又は arg_1 , arg_2 ; mM	ステップ、トレース、ブレーク・ポイントの各コマンド入力前に表示モードを指定できる。 n（m）　表　示 0　PC　AF 1　全レジスタ 2（0）　PC　AFとメモリ・ブロック 3（1）　全レジスタとメモリ・ブロック 但し arg_1　メモリ・ブロックの先頭 arg_2　メモリ・ブロックの最後

ユーザ・レジスタの表示、変更コマンドにおいても、ターミネータをメモリ内容の表示、変更コマンドと同じように使用できる。この場合のメモリ・アドレスは確定している。表4.1参照。

メモリ内容、及び　ユーザ・レジスタの表示、変更コマンド以外のモニタ・コマンドはターミネータを持たない。これらのコマンドでは、コマンド記号を入力することによって実行を開始し、実行完了後、次のコマンド待ちの状態となる。

〔例〕　＊１２３４；B

＊　　　　　…………次のコマンド待ち

5. 操 作 方 法

5－1　前 準 備

5－1－1　動作モード

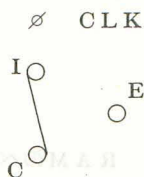
出荷時において本ボードの動作モードは以下のように設定されている。下記の動作モードと異なるモードで使用する場合、3. 4動作モードの選択に従って設定を変えなければならない。

ラッピング端子

① システム・クロック

ø　CLK

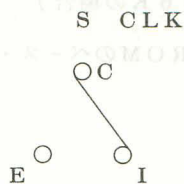
I　E

C

（説明）

出荷時は内部クロック側（Ｉ－Ｃ）に結線されている。

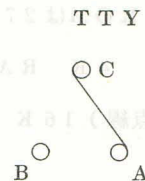
外部クロックを使用する場合、Ｅ－Ｃを結線する。

② シリアル・クロック

S　CLK

（説明）

出荷時は内部シリアル・クロック側（Ｉ－Ｃ）に結線されている。外部シリアル・クロックを使用する場合、Ｅ－Ｃを結線する。

③ ＣＴＳ接地

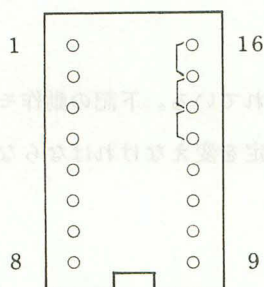
ＴＴＹ

（説明）

出荷時はＡ－Ｃを結線している（ＴＴＹ使用時はこの結線を使用しなければならない）。

その他の場合はＡ－Ｂを結線する。

Ｂ－Ｃを結線してはならない。

プラットホーム端子

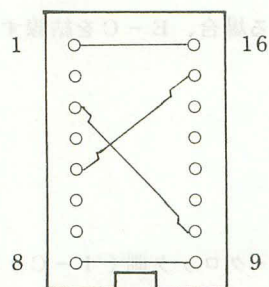
① K_1

（説明）

13−16、13−15、13−14 結線　８２５１より割り込み　なし

4−13、 5−12、 6−11 開放　ボー・レート　１１０

7−10　開放　ＥＰＲＯＭ ＃３、＃４ 汎用使用しない

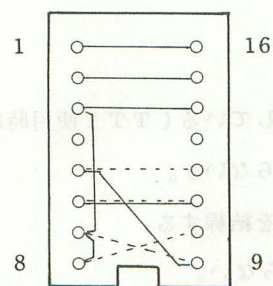
（注）　ＴＴＹ使用時　ボー・レート　１１０

② K_2

（説明）

1−16、5−15 結線　ユーザ　ＲＡＭのベース・アドレス
（ ４Ｋの場合 ）　００００

5−15−16　結線　ユーザ　ＲＡＭのベース・アドレス
（１６Ｋの場合）　００００

8−9、3−10　結線　ＯＳ　ＲＯＭのベース・アドレス
Ｅ０００

③ K_3

（説明）

1−16、2−15、3−14 結線　ＯＳ　ＲＯＭは２７０８相当

6−11、5−9 7−8−3 結線（実線）　４Ｋ　ＲＡＭ

又は

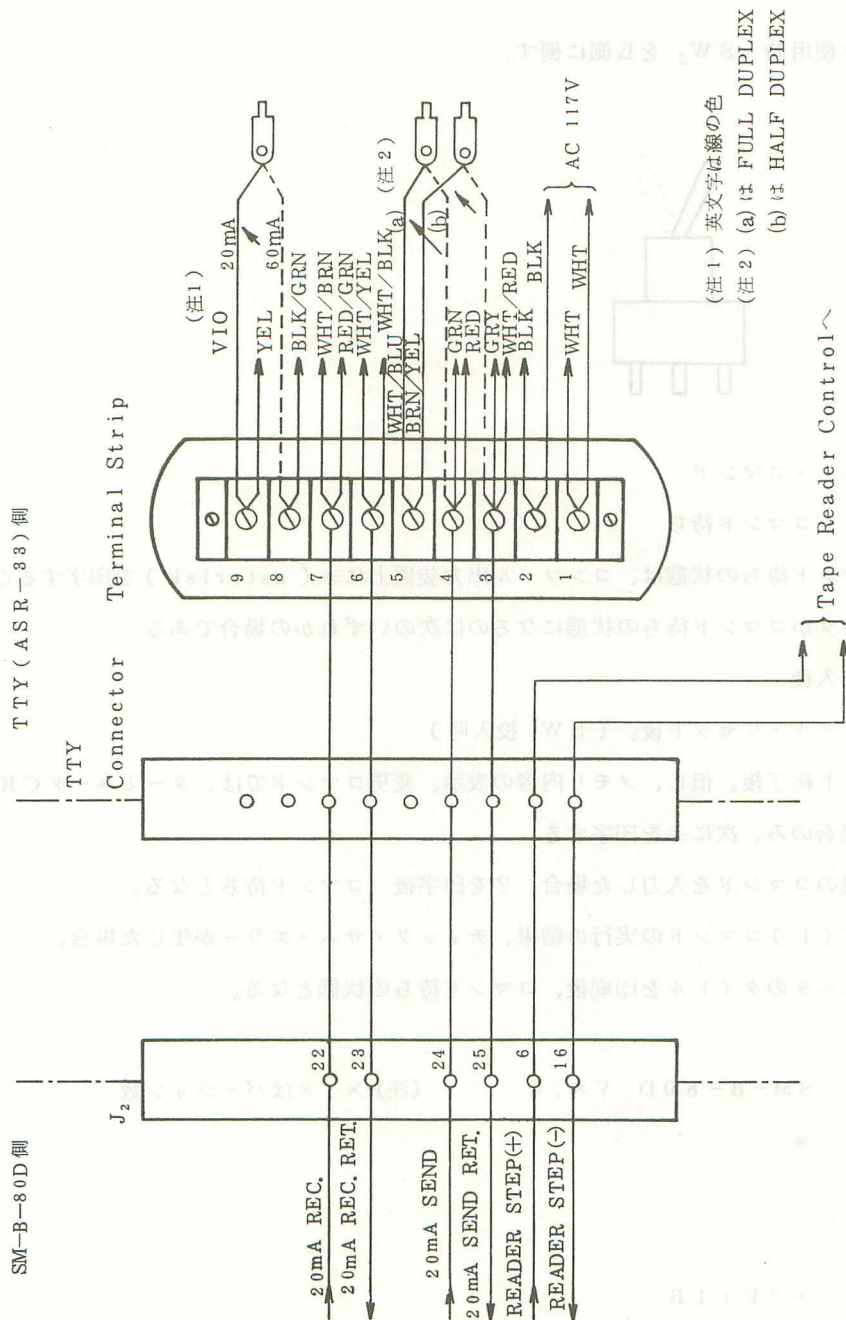
5−12、6−11、7−9、8−10 結線（点線）１６Ｋ　ＲＡＭ

５−１−２　$^I/_O$ 装置

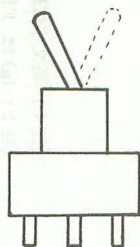
システム $^I/_O$ 装置としてＴＴＹを使用する場合の結線図を図５．１に示す。

図5．1　TTY 結線法

（注1）英文字は線の色
（注2）(a) は FULL DUPLEX
　　　 (b) は HALF DUPLEX

（注3）TTY は20mAループ電流，かつ FULL DUPLEX にすること。
（点線から実線に変更する）

5－1－3　その他

スイッチ

モニタ使用時 SW_2 をE側に倒す。

5－2　モニタ・コマンド

5－2－1　コマンド待ち

モニタのコマンド待ちの状態は、コンソール出力装置上に＊（asterisk）を印字することにより示す。モニタがコマンド待ちの状態になるのは次のいずれかの場合である。

(1) 電源投入後。

(2) マニュアル・リセット後。（SW_1 投入時）

(3) コマンド終了後。但し、メモリ内容の表示、変更コマンドでは、ターミネータCRを入力した場合のみ、次に＊を印字する。

(4) 未定義のコマンドを入力した場合、？を印字後 コマンド待ちとなる。

(5) ロード（L）コマンドの実行の結果、チェック・サム・エラーが生じた場合。

(1)、(2)では、モニタのタイトルを印刷後、コマンド待ちの状態となる。

(1)、(2)の例

```
SM-B-80D V×.×        (注) ×.×はバージョン数
＊
```

(3)の例

```
＊2F；1B
＊
```

(4)の例

＊；H？

＊

(5)の例　　＊；L

０Ｅ３Ｆ

＊

５－２－２　　プログラムのロード

形　式

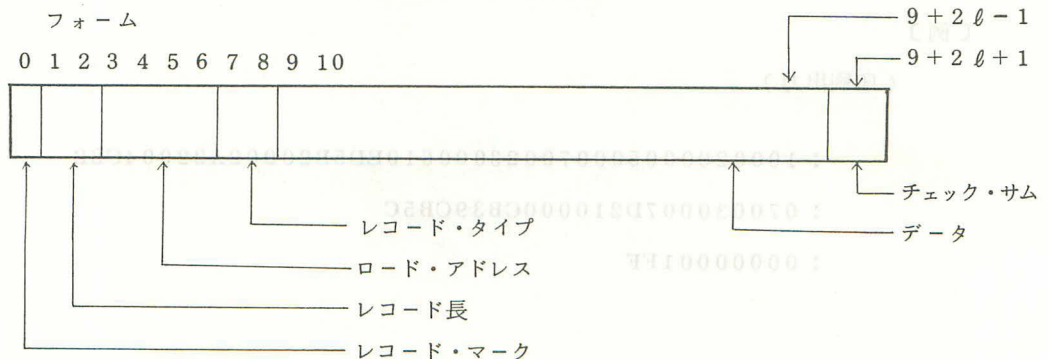
＊；Ｌ　（ 〜〜 はキー入力する部分、以下同じ）

オブジェクト入力チャンネル（ＯＩ）に指定された入力装置よりオブジェクト・プログラムを内部メモリにロードする。オブジェクト・プログラムはインテル標準１６進フォーマットでなければならない。本ボードでは、特にＯＩを指定しない場合入力装置として紙テープ・リーダが選ばれる。

〔例〕

＊；Ｌ

＊

インテル標準１６進フォーマットは次の形式をしている。

(1)　レコード・マーク

コロン（：）に対応するＡＳＣＩＩコード　３Ａによりレコードの先頭を示す。

(2) レコード長

レコードのデータ（バイト）長をＡＳＣＩＩコードにより表わす。例えば、データ長１２９バイトは１６進数で８１であり、ＡＳＣＩＩコードでは３８３１となる。エンド・オブ・ファイルのレコード長は００であり、フレーム１、２には対応するＡＳＣＩＩコードが入る。

(3) ロード・アドレス

レコードの先頭のデータがロードされるアドレスをＡＳＣＩＩコードにより表わす。アドレスは４桁の１６進数で表わされ、上位側がフレーム３、４に入る。エンド・オフ・ファイルのアドレスは００００であり、フレーム３〜６には対応するＡＳＣＩＩコードが入る。

(4) レコード・タイプ

データ・レコードはタイプ０、エンド・オブ・ファイルはタイプ１とし、各々０、１に対応するＡＳＣＩＩコードで表わす。

(5) データ

１バイト・データは２桁の１６進数で表現できるが、これを対応するＡＳＣＩＩコードで連続する２フレームに入れる。データの上位バイトが前のフレーム、下位バイトが後のフレームとなる。エンド・オブ・ファイルにはデータはない。

(6) チェック・サム

第１フレームからデータの最終フレームまでの数を２フレーム毎に区切り、各フレームのデータ（ＡＳＣＩＩコードではない）を２進数表現にする。これらの各単位の２進数を２進加算し、結果の２の補数をチェック・サムとして２桁の１６進数で表わす。チェック・サムは対応するＡＳＣＩＩコードにより最後の２フレームに入る。

〔例〕

（印刷出力）

```
:10002000050007002300061 0ED5B20002A22004C8B
:070030007D210000CB39CB5C
:00000001FF
```

5－2－3　　プログラムのパンチ

形　式

＊　arg1、arg2 ；P

arg1、arg2 で指定されるメモリ・アドレスの内容をオブジェクト出力チャンネル（００）で、指定された出力装置へ出力する。出力形式はインテル標準１６進フォマットである。本ボードでは、特に００を指定しない場合、出力チャンネルとしてＴＴＹの紙テープ・パンチャが選ばれる。

〔例〕

```
＊２６，４７；Ｐ
:10002600061OED5B20002A22004C7D210000CB3912
:10003600CB1F300119CB23CB1210F3222400C31699
:02004600E100D7
:00000001FF
＊
```

5－2－4　　メモリ・レジスタの内容表示

形　式

＊　arg1／nn

arg1 でメモリ・アドレスかユーザ（ＣＰＵ）レジスタを指定する／入力によって指定されたアドレスかレジスタの内容を１６進数 nn で表示する。 表示後、↓（ＣＲ　carridge return）、ＬＦ（ line feed ）、↑（ up arrow ）のいずれかのターミネータ待ちの状態にある。↓を入力すると現在のコマンドは完了し、次のコマンド待ちの状態となる。ＬＦを入力すると現在のメモリ・アドレスを＋１して、さらにメモリ内容の表示を行い、次のコマンド待ちの状態となる。↑を入力すると現在のメモリ・アドレスを－１して、さらにメモリ内容の表示を行い、次のコマンド待ちの状態となる。

〔例〕

```
＊２０／０５
＊
```

（紙テープ）

データ・ファイル
フレーム
0
1
2
3
4
5
6
7
8
9
10
11
12
39
40
41
42
3 A :
3 1 1
3 0 0
3 0 0
3 0 0
3 2 2
3 0 0
3 0 0
3 0 0
3 0 0
3 5 5
3 0 0
3 0 0
3 4 4
4 3 C
3 8 8
4 2 B
レコード・マーク
レコード長
ロード・アドレス
レコード・タイプ
データ
チェック・サム
印刷形式
紙テープ（ASCII表現）
エンド・オブ・ファイル
3 A :
3 0 0
3 0 0
3 0 0
3 0 0
3 0 0
3 0 0
3 0 0
3 1 1
4 6 F
4 6 F
レコード・マーク
レコード長
ロード・アドレス
レコード・タイプ
チェック・サム

＊２０／０５
００２１／００
００２２／０７
００２３／００
００２４／２３
＊

＊２４／２３＾
００２３／００＾
００２２／０７＾
００２１／００＾
００２０／０５
＊

＊．ＰＣ／００２Ｆ
＊．Ａ／７０
＊．Ｈ'／ＤＡ
＊．ＩＸ／５Ｅ５Ｆ
＊．ＩＹ／Ｆ８５８

5－2－5　　メモリ・レジスタの内容変更

形　式

＊arg1 ／ nn mm

既述のメモリ・レジスタの内容表示コマンドによりnnを表示後、ターミネータを入力せずに１６進数mm を入力するとarg1 で示されるアドレスかユーザ・レジスタの内容は mm に変更される。mm 入力後ターミネータ待ちの状態になる。この場合のターミネータの意味は既述のメモリ、レジスタの内容表示の場合と同じである。３桁以上の１６進数を入力すると最後の２桁が有効になる。

〔例〕

```
＊２０／０５　Ｆ６
００２１／００　２Ａ
＊２０／Ｆ６
００２１／２Ａ
＊
```

```
＊．ＰＣ／００２Ｆ　３２
＊．Ａ／７０　３Ｂ
＊．Ｈ'／ＤＡ　ＦＦ
＊．ＩＸ／５Ｅ５Ｆ　５Ｅ８０
＊．１Ｙ／Ｆ８５８　５Ｅ６０
```

5－2－6　メモリ・ブロックの表示

形　式

＊arg1 ，arg2 ／

arg1 、arg2 で指定されるメモリ・アドレスの範囲の内容を２桁の１６進数で印刷出力する。出力後次のコマンド待ちとなる。

〔例〕

```
*FF00,FFFF/
FF00/B0 E5 AF FF 00 00 AF A5 A1 A0 02 00 FF FF FF 00
FF10/00 00 2F A5 A1 E1 AF 0F A0 A4 2F A7 E4 B0 AF 25
FF20/A5 E5 AF AD A0 A5 2F AF A4 A4 AF 27 A1 A0 AF 25
FF30/A0 25 4B E7 5E E7 49 E7 5E E7 49 E7 5E E7 54 45
FF40/80 A1 AF AF E0 E4 AF 27 A0 80 AF 07 E1 E4 8D 2D
FF50/A1 A5 8F A7 A4 A4 2F 27 A0 F4 27 AD A0 B0 AF 85
FF60/A0 A5 AF AF A0 B0 AF 8D A0 E0 AD 0F B0 A0 A7 2F
FF70/A0 A1 AF 8D A0 A1 AD 8D A4 B1 8F AF E0 E0 AF 07
FF80/50 50 5F 5F 70 5A 1F 5E 50 5A 0F 5B 70 52 5F 5B
FF90/5A 52 5F 4E 50 7A 5F 4E 52 5A 5F 4E 50 58 5F 5A
FFA0/50 52 5F 5F 72 50 5F 5F 50 7A 4F 5A 50 70 5F 5B
FFB0/50 52 5F 5F 50 D0 5F 5E 50 D0 4F 1B 50 50 5F 5F
FFC0/52 58 0F 5E 50 58 5F 4F 5A D8 5F 5F 70 58 5F 5F
FFD0/5A 52 0E 5F 50 D2 1F 5E 64 E0 64 E0 DB FF 64 E0
FFE0/E0 FF A0 E4 16 E1 B4 FF 58 F8 5B 5E 50 D2 5E 5E
FFF0/DA FA 5F 1F 50 50 5F 5F D8 D0 5F 5F 52 70 1B 5F
*
```

５－２－７　ユーザ・レジスタの表示

形　式

＊；R　または　＊；１R

このコマンドはすべてのユーザ（ＣＰＵ）レジスタの内容を表示する場合に使用する。特定のユーザ・レジスタの表示はメモリ内容の表示コマンドにおいてその arg_1 にユーザ・レジスタに対応するニーモニック（例えば　・ＰＣ）を用いることによって行うことができる。

；Rはレジスタの内容だけを表示する場合に使用し、；１Rはレジスタの内容と対応するレジスタのラベルを表示する場合に用いる。

〔例〕

```
＊２F；１B
＊２６；S
００２８　７２７２
＊；R
0028 7272 5F01 10D8 0005 0007 1F4F 7ADA 5E4E DA52 5E5F F858 FFB0
＊；１R
 PC   AF   I,IF  BC   DE   HL   AF'  BC'  DE'  HL'  IX   IY   SP
0028 7272 5F01 10D8 0005 0007 1F4F 7ADA 5E4E DA52 5E5F F858 FFB0
＊
```

５－２－８　ユーザ・プログラムの実行

形　式

＊arg_1　；G

arg_1 で指定されるメモリ・アドレスからプログラム（ユーザ・プログラム）を実行する。arg_1 を省略した場合、現在のＰＣ（ユーザＣＰＵレジスタで・ＰＣの内容に等しい）で示されるアドレスからプログラムを実行する。

〔例〕

```
＊２６；G
＊

＊．ＰＣ／Ｅ０５Ｆ　２６
＊；G
＊
```

5－2－9　ブレーク・ポイントの設定と解除

形　式

＊arg1 ; nB

arg1 で指定されるメモリ・アドレスにブレーク・ポイントを設定する。nはブレーク・ポイントの識別番号で０～７の整数であり、０の場合は省略可能である。同じメモリ・アドレスにブレーク・ポイントを２回以上設定してはならない。arg1 を省略すると識別番号nのブレーク・ポイントを解除する。

ブレーク・ポイントの動作は、プログラム実行を中断する（ break　ブレーク）アドレスに特定の出力命令（ＯＵＴ　（ＤＥ），Ａ）を挿入することにより実現している。したがって、ブレーク・ポイントは必ず命令の先頭アドレスに設定しなければならない。またこの出力命令は２バイト命令であるので例２のような場所には設定してはならない。

プログラム実行の結果、ブレーク・ポイント動作が成立すると、後述する表示モードの指定に従い、ブレーク・ポイント直前の実行状態と／を印刷し、実行を中断してモニタ・コマンド待ちとなる。この場合、対応するブレーク・ポイントは解除される。

〔例１〕

```
＊０２Ｃ；１Ｂ
＊３Ａ；５Ｂ
＊０００３０；２Ｂ
＊；Ｄ
１　００２Ｃ
２　００３０
５　００３Ａ
＊；Ｋ
＊；Ｄ
＊
```

〔例２〕

```
　　　　　ＪＲ　　　ＮＣ，ＬＡＢＥＬ　　　　　　　(注)
　　　　　ＬＤ　　　Ａ，Ｂ　　　←－－－－－－－－－　矢印のアドレスに設定
ＬＡＢＥＬ：ＡＤＤ　　Ａ，Ｃ　　　　　　　　　　　　　してはならない。
```

5－2－10　ブレーク・ポイントの表示

形　式

＊　;D

現在設定されているすべてのブレーク・ポイントの識別番号nとそのメモリ・アドレスをnの順に表示する。

〔例〕

＊2C；1B

＊3A；5B

＊38；7B

＊30；2B

＊00036；4B

＊；D

1　002C

2　0030

4　0036

5　003A

7　0038

＊

5－2－11　　ブレーク・ポイントの全解除

形　式

＊　；K

現在設定されているすべてのブレーク・ポイントを解除する、なお、特定のブレーク・ポイントの解除は次のいずれかによって行うことができる。

(1) コマンド入力；nBを入力した場合。

(2) 命令実行の結果、特定のブレーク・ポイントが成立した場合。

〔例〕

```
＊2C；1B
＊30；2B
＊3A；5B
＊；D
1  002C
2  0030
5  003A
＊；K
＊；D
＊


＊02C；1B
＊3A；5B
＊00030；2B
＊；D
1  002C
2  0030
5  003A
＊26；G
002C  72F0  1／
＊；D
2  0030
5  003A
＊；5B
＊；D
2  0030
＊
```

5－2－12　　ステップ

形　式

＊arg1 ; nS

arg1で指定されるメモリ・アドレスよりn命令実行し、各命令実行毎にその結果を後述する表示モードの指定に従って印刷出力する。nは０～２５５ の整数であり、nを0とするか省略した場合n＝１と等価である。n命令実行中にブレーク・ポイントがある場合、そのブレーク・ポイントで命令実行は中断し、結果を印刷して次のコマンド待ちとなるので、その時点で残りのステップ動作は無効となる。arg1 を省略した場合、現在のＰＣ（ユーザＣＰＵレジスタ）からステップ動作を行う。

〔例〕

```
＊02C；1B
＊3A；5B
＊00030；2B
＊26；S
0028　72F0
＊；S
002C　72F0　1/
＊；S
002F　72F0
＊；S
0030　72F0
＊；S
0030　72F0　2/
＊
```

```
＊2C；1B
＊30；2B
＊3A；5B
＊26；5S
0028　3244
002C　3244
002C　3244　1/
＊
```

５－２－１３　　トレース

形　式

arg_1 ；Ｔ

arg_1 で指定されるメモリ・アドレスよりプログラムを実行し、各命令実行毎に後述の表示モードの指定に従って実行結果を印刷出力する。本コマンドは、ブレーク・ポイント成立か、ＣＴＲＬ　Ｃ　のキー入力によって以後無効となり、次のコマンド待ちとなる。arg_1 を省略した場合、現在のＰＣ（ユーザＣＰＵレジスタで　・ＰＣの内容）の値よりトレース動作を行う。

〔例〕

```
＊２Ｃ；１Ｂ
＊３０；２Ｂ
＊３Ａ；５Ｂ
＊２６；Ｔ
００２８　７２Ｆ０
００２Ｃ　７２Ｆ０
００２Ｃ　７２Ｆ０　１／
＊

＊．ＰＣ／００２Ｃ　２６
＊；Ｔ
００２８　７２Ｆ０
００２Ｃ　７２Ｆ０
００２Ｆ　７２Ｆ０
００３０　７２Ｆ０
００３０　７２Ｆ０　２／
＊
```

5－2－14　表示モードの指定

形　式

①　＊arg1 ，arg2 ；nM　　　arg1 ，arg2 で指定されるメモリ・ブロックレジスタ

②　＊；nM　　　既に指定されているメモリ・ブロックとレジスタ

本コマンドは、ブレーク・ポインド、ステップ、トレースの各コマンドにおいて、印刷出力のモード指定を行う場合に用いる。①のコマンドはメモリ・ブロックの範囲を指定するために用いられ、nによって表示レジスタの種類を指定する。②のコマンドにおいては、メモリ・ブロックの範囲は既に指定されている値に等しく、またnによって、メモリ・ブロックの印刷出力の有無と表示レジスタの種類を指定する。

なお、arg1 、arg2 は各々メモリ・ブロックの先頭と最後のアドレスを示す。

コ　マ　ン　ド	n	説　　　明
＊arg1 ，arg2 ；nM	0か省略	PC，AFとarg1 ，arg2 で指定されるメモリ・ブロック
	1	全レジスタとarg1 ，arg2 で指定されるメモリ・ブロック
＊；nM	0か省略	PC、AFのみ表示
	1	全レジスタを表示
	2	PC、AFと既に指定されているメモリ・ブロック
	3	全レジスタと既に指定されているメモリ・ブロック

〔例1〕

```
*2C;1B
*30;2B
*36;5B
*38;7B
*20,3F;0M
*26;T
0028 72F2
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 D3 DE 00 4C
0030/ D3 DE 00 00 CB 39 D3 DE D3 DE 19 CB 23 CB 12 10

002C 72F2
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 D3 DE 00 4C
0030/ D3 DE 00 00 CB 39 D3 DE D3 DE 19 CB 23 CB 12 10

002C 72F2 1/
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ D3 DE 00 00 CB 39 D3 DE D3 DE 19 CB 23 CB 12 10

*20,3F;1M
*2C;T
002F 72F2 5F01 10D8 0005 0007 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ D3 DE 00 00 CB 39 D3 DE D3 DE 19 CB 23 CB 12 10

0030 72F2 5F01 1000 0005 0007 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ D3 DE 00 00 CB 39 D3 DE D3 DE 19 CB 23 CB 12 10

0030 72F2 5F01 1000 0005 0007 5F5F FADA 5F5E DA52 5F5B 5070 FFB4 2/
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 D3 DE D3 DE 19 CB 23 CB 12 10

*3A;3B
*3D;4B
*;1R
 PC   AF  1,1F  BC   DE   HL   AF'  BC'  DE'  HL'  IX   IY   SP
0030 72F2 5F01 1000 0005 0007 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
*30;T
0031 07F2 5F01 1000 0005 0007 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 D3 DE D3 DE D3 DE 23 D3 DE 10

0034 07F2 5F01 1000 0005 0000 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 D3 DE D3 DE D3 DE 23 D3 DE 10

0036 0744 5F01 1000 0005 0000 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 D3 DE D3 DE D3 DE 23 D3 DE 10

0036 0744 5F01 1000 0005 0000 5F5F FADA 5F5E DA52 5F5B 5070 FFB4 5/
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 CB 1F D3 DE D3 DE 23 D3 DE 10
```

〔例２〕

arg1 ，arg2は前の設定値が残っている。

```
*;0M
*36;T
0038 0305
0038 0305 7/
*;1M
*38;T
003A 0305 5F01 1000 0005 0000 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
003A 0305 5F01 1000 0005 0000 5F5F FADA 5F5E DA52 5F5B 5070 FFB4 3/
*;2M
*3A;T
003B 0304
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 CB 1F 30 01 19 CB 23 D3 DE 10

003D 030C
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 CB 1F 30 01 19 CB 23 D3 DE 10

003D 030C 4/
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 CB 1F 30 01 19 CB 23 CB 12 10

*3F;1B
*;3M
*3D;T
003F 0344 5F01 1000 000A 0005 5F5F FADA 5F5E DA52 5F5B 5070 FFB4
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 CB 1F 30 01 19 CB 23 CB 12 D3

003F 0344 5F01 1000 000A 0005 5F5F FADA 5F5E DA52 5F5B 5070 FFB4 1/
0020/ 05 00 07 00 23 00 06 10 ED 5B 20 00 2A 22 00 4C
0030/ 7D 21 00 00 CB 39 CB 1F 30 01 19 CB 23 CB 12 10

*
```

5－2－15　　ポート入力

形　式

＊arg1 ；nI

＊；nI

arg1 ＋nで指定されるポートからデータ（1バイト）を読み込み表示する。nを省略した場合はn＝0と等価である。nは 0～255 の10進数とする。また、arg1 を省略した場合は、以前に指定されたarg1 が有効となりarg1 ＋nが指定される。

〔例〕

＊1A；1I
3B
＊；1I
3B
＊A0；I
FF
＊；I
FF
＊

5－2－16　　ポート出力

形　式

＊arg1 ，arg2；nO

＊arg1 ；nO

arg1 ＋nで指定されるポートへarg2 で示される1バイト データを書き込む。nを省略した場合は、n＝0と等価である。nは0～255の10進数とする。また、arg2 を省略した場合は、以前に指定されたarg1 が有効となり、ポートarg1 ＋nへarg1 （現在のコマンド ライン）を書き込む。

〔例〕

＊B0，FE；2O
＊FB；2O
＊3B，FA；O
＊FC；O
＊

6. 規　　　格

〈一般規格〉

表6.1　一般規格

項　目	規　格	備　考
C P U	Z－80　CPUチップ	
語　長	1語　8ビット 命令　8，16，24，32ビット データ　8ビット アドレス　16ビット I/Oアドレス　入力，出力，各8ビット	
最小命令実行時間	1.63μs	4クロック・サイクル 8ビット　レジスタ加算 8ビット　レジスタ間転送
CPUクロック	内部クロック（水晶発振）　2.4576 MHz　または 外部クロック　0.7～2.5 MHz	下限周波数はダイナミックRAMのリフレッシュ・サイクルで決まる。
メ　モ　リ	OS　ROM　LH－2708　最大4個実装可能 OS　RAM　LH－2111A4　2個 ユーザRAM　LH－4027－3　またはLH－4116－3　8個	ソケット実装 ソケット実装
メモリ容量	OS　ROM　最大4Kバイト OS　RAM　256バイト ユーザRAM　4Kバイト　または　16Kバイト	ユーザRAM 4Kバイト実装製品　LH－8H01A 16Kバイト実装製品　LH－8H01B
メモリ・アドレス	OS　ROM　4Kバイト単位にベース・アドレス設定可能 OS　RAM　FF00～FFFF（固定） ユーザRAM　4Kバイト　または　16Kバイト単位にベース・アドレス設定可能	ジャンパ端子　K_2 ジャンパ端子　K_2
パラレル I/O インターフェイス	PIO　1個使用 8ビット　入出力データ線　×2 2ビット　シェイク・ハンド制御線　×2 ユーザ配線領域　16ピンDIP　IC　4個実装可能 コネクタ　50ピン　フラット・ケーブル用	コネクタJ_1
シリアル I/O インターフェイス	UART（8251）　1個使用 TTYインターフェース（20mA電流ループ）及び RS－232C インターフェース コネクタ　26ピン　フラット・ケーブル用	コネクタJ_2

項　　目	規　　　　格	備　　考
カウンタ／タイマ	ＣＴＣ　１個使用 チャンネル０　　システム使用（ボー・レート作成用） チャンネル１〜３　ユーザ開放 クロック入力　　2.4576MHz（406.9 ns　）	
I/O ポート アドレス	ユーザ開放　　００〜ＣＦ システム使用　Ｄ０〜ＤＦ システム・リザーブ　　Ｅ０〜ＦＦ 但し　Ｄ０　ＰＩＯ　ポートＡ　データ Ｄ１　ＰＩＯ　ポートＡ　コントロール Ｄ２　ＰＩＯ　ポートＢ　データ Ｄ３　ＰＩＯ　ポートＢ　コントロール Ｄ８　ＣＴＣ　チャンネル　０ Ｄ９　ＣＴＣ　チャンネル　１ ＤＡ　ＣＴＣ　チャンネル　２ ＤＢ　ＣＴＣ　チャンネル　３ ＤＣ　ＵＡＲＴ　データ ＤＤ　ＵＡＲＴ　コントロール ＤＥ　Ｗ　システム　ＮＭＩ（Ｎ－Delay） Ｒ　ボー・レート、アドレスＥ　リセット ＤＦ　Ｗ　システム　ＮＭＩ（Delay） Ｒ　リーダ・ステップ	ブレイク・ポイント ステップ・トレース
ボー・レート	８種類　切り換え　可能 110,150,300,600,1200,2400,4800,9600	ジャンパ端子　K_1
電　　　源	＋５Ｖ±５％　　2.2 Amax ＋１２Ｖ±５％　　４５０mA　max －１２Ｖ±５％　　１５０mA　max	
動作温度	０℃　〜　５０℃	
ボード寸法	２７０×１９０×２０　　単位　mm	
	J_1　５０ピン　フラット・ケーブル・コネクタ （ヒロセ　ＨＩＦ３－５０Ｐ－2.54ＤＳ相当） J_2　２６ピン （同　上　ＨＩＦ３－２６Ｐ－2.54ＤＳ相当） J_3　１００ピン　コネクタ　3.175mm　ピッチ （ケル製　４８００－１００－１３５　相当）	

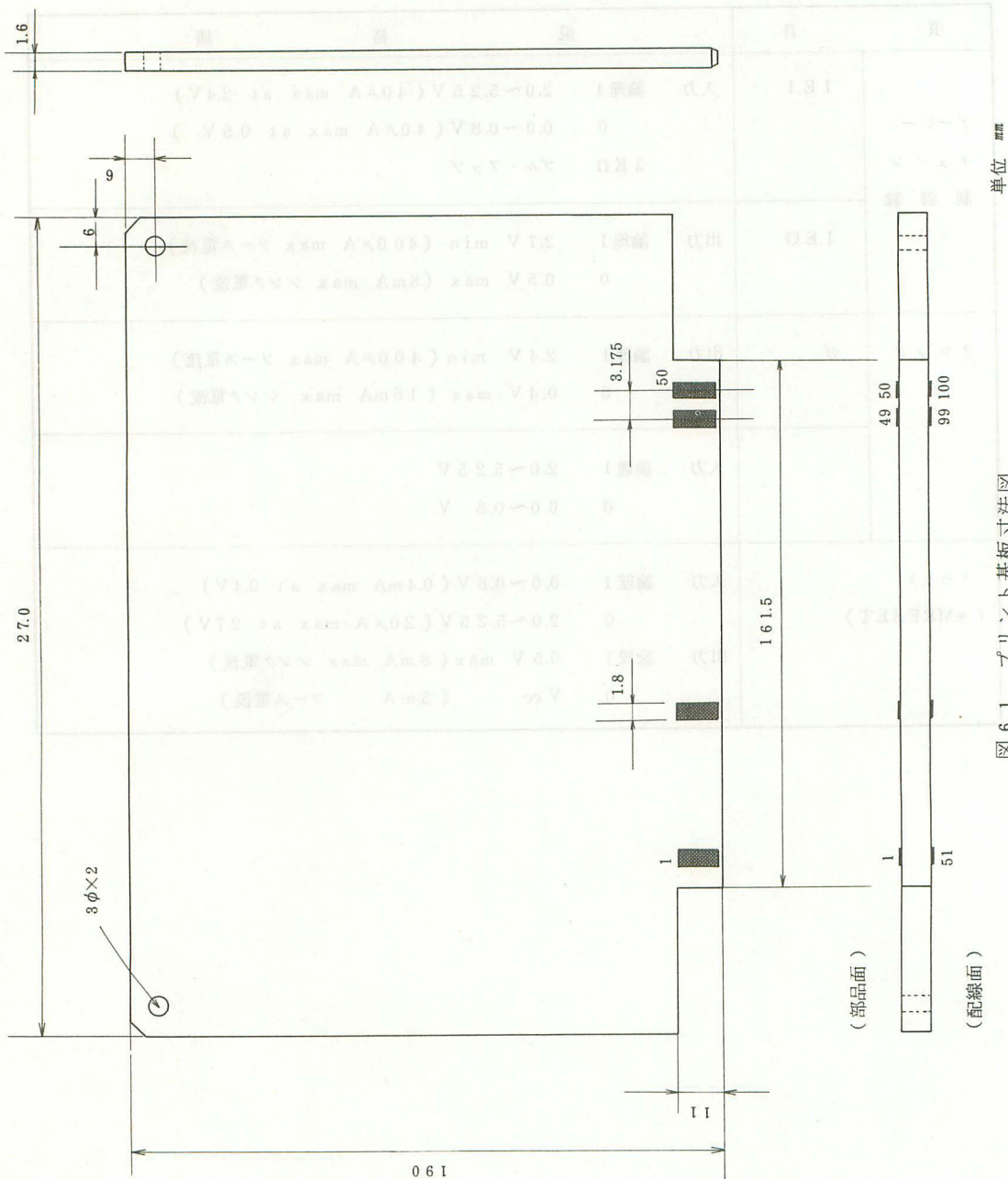
〈バス信号線〉

表6.2　バス信号線　規格

項　　目		規　　格　　値
アドレス・バス (＊A_0〜＊A_{15}) データ・バス (＊D_0〜＊D_7)		トライ・ステート　　TTLコンパティブル 入力　論理1　0.0〜0.8V(200μA max at 0.4V) 　　　　0　2.2〜5.25V(25μA max at 5.25V) 出力　論理1　0.5Vmax (40mA シンク電流) 　　　　　　2.6Vmin (10mA ソース電流) 出力オフ状態リーク電流　100μA max (Vcc=5.25V, Vo=$^{5.25}/_{0.4}$V)
入出力制御線 ＊RFSH, ＊MREQ ＊IORQ, ＊RD ＊WR, ＊M_1		トライ・ステート　　TTLコンパティブル 入力　論理1　0.0〜0.8V(40μA max at 0.5V) 　　　　0　2.0〜5.25V(40μA max at 2.4V) 出力　論理1　0.4V max (32mA シンク電流) 　　　　0　2.4V min (5.2mA ソース電流) 出力オフ状態リーク電流　40μA max(Vcc=5.25V, V_0=$^{2.4}/_{0.4}$V)
入　力 制 御 線	＊INTU ＊NMIU	入力　論理1　0.0〜0.8V 　　　　0　2.0〜5.25V 　　　3KΩ　プル・アップ
	＊WAIT ＊BUSRQ ＊RENB	入力　論理1　0.0〜0.8V(40μA max at 0.5V) 　　　　0　2.0〜5.25V(40μA max at 2.4V) 　　　3KΩ　プル・アップ
	＊DEBG	入力　論理1　0.0〜0.8V(0.8mA max at 0.4V) 　　　　0　2.0〜5.25V(40μA max at 2.7V)
	＊DDIS	入力　論理1　0.0〜0.8V(200μA max at 0.4V) 　　　　0　2.0〜5.25V(25μA max at 5.25V) 　　　1KΩ　プル・アップ(内部で74LS03とwired−0Rになっている)
出　力 制 御 線	＊BUSAK	出力　論理1　0.4V max (16mA シンク電流) 　　　　0　2.4V min (400μA ソース電流)
	＊HALT	出力　論理1　0.4V max (100mA シンク電流) 　　　　0　Vcc (15mA typ ソース電流)

項目		規格値
デージー・チェイン制御線	IEI	入力　論理1　2.0～5.25V(40μA max at 2.4V) 　　　0　0.0～0.8V(40μA max at 0.5V) 　　　3KΩ　プル・アップ
	IEO	出力　論理1　2.7V min (400μA max ソース電流) 　　　0　0.5V max (8mA max シンク電流)
クロック	∅	出力　論理1　2.4V min(400μA max ソース電流) 　　　0　0.4V max (16mA max シンク電流)
		入力　論理1　2.0～5.25V 　　　0　0.0～0.8 V
リセット(＊MRESET)		入力　論理1　0.0～0.8V(0.4mA max at 0.4V) 　　　0　2.0～5.25V(20μA max at 2.7V) 出力　論理1　0.5V max(8mA max シンク電流) 　　　0　Vcc (5mA ソース電流)

図 6.1 プリント基板寸法図

単位 mm

付　録　A

ＳＭ－Ｂ－８０Ｄ　モニタ　ユーザ開放サブルーチン

ＳＭ－Ｂ－８０Ｄ のモニタを使用した場合、ユーザ・プログラム内で次の１２種のサブルーチンを使用できる。

	エントリ・アドレス	サブルーチン名	
1.	Ｅ０４６	ＲＥＡＤ	
2.	Ｅ０４Ｂ	ＷＲＩＴＥ	
3.	Ｅ７２７	ＳＰＡＣＥ	
4.	Ｅ７１９	ＥＣＨＯ	
5.	Ｅ７１Ｅ	ＣＲＬＦ	
6.	Ｅ７０Ｅ	ＰＲＨＥＸ	
7.	Ｅ５４Ａ	ＰＲＡＤＲ	
8.	Ｅ７０６	Ａ２ＢＩＮ	
9.	Ｅ７３７	ＰＲＭＥＳ	
10.	Ｅ１１Ｃ	ＥＸＩＴ	
11.	Ｅ３１８	ＦＥＥＤＥＲ	
12.	Ｅ５５Ｆ	ＧＥＴＡＲＧ	（ＡＲＧＢＵＦ　ＦＦＯＢ）

一般的使用法

ＣＡＬＬ　　エントリ・アドレス

又は

サブルーチン名　ＥＱＵ　エントリ・アドレス

ＣＡＬＬ　　サブルーチン名

1. ＲＥＡＤ

指定チャンネルＣＨＡＮＥＬより１キャラクタを読み込む。読み込んだキャラクタは、ＡＳＣＩＩコードに変換されてＡ、Ｄレジスタに格納される。Ａ、Ｄは同一内容。CHANEL についてはＡ－５ページ参照のこと。

```
LD      E,CHANEL
CALL    READ
```

注　Ｅの内容は不変

Ａ、Ｆ、Ｄの内容は変化する。

2. ＷＲＩＴＥ

指定チャンネルＣＨＡＮＥＬへＤレジスタの内容ＣＨＡＲ１を出力する。

```
LD      E,CHANEL
LD      D,CHAR1
CALL    WRITE
```

注　Ｅ，Ｄの内容は不変。

Ａ，Ｆの内容は変化する。

3. ＳＰＡＣＥ

指定チャンネルCHANELへスペース（ＡＳＣＩＩ　２０Ｈ）を出力する。

```
LD      E,CHANEL
CALL    SPACE
```

注　Ｅの内容は不変。

Ａ，Ｆ，Ｄの内容は変化する。

4. ＥＣＨＯ

指定チャンネルCHANELより１キャラクタをＡ，Ｄレジスタへ読み込み、同一チャンネルへエコー・バックする。

```
LD      E,CHANEL
CALL    ECHO
```

注　Ｅの内容は不変。

Ａ，Ｆ，Ｄの内容は変化する。

これは次と同じ操作である。

```
LD      E,CHANL
CALL    READ
```

```
    CALL        WRITE
```

5. CRLF

指定チャンネルCHANELへCR，LFを出力する。

```
    LD      E，CHANEL
    CALL        CRLF
```

注　Eの内容は不変。

A，F，Dの内容は変化する。

これは次と同じ操作である。

```
    LD      D，0DH
    CALL        WRITE
    LD      D，0AH
    CALL        WRITE
```

6. PRHEX

Aレジスタの内容を2桁の16進数に変換し、指定チャンネルCHANELへ出力する。

```
    LD      E，CHANEL
    LD      A，DATA
    CALL        PRHEX
```

注　Eの内容は不変。

A，F，Dの内容は変化する。

7. PRADR

HLレジスタの内容を4桁の16進数に変換し、指定チャンネルCHANELへ出力する。

```
    LD      E，CHANEL
    LD      HL，ADDRS
    CALL        PRADR
```

注　Eの内容は不変。

A，F，Dの内容は変化する。

8. ASBIN

Aレジスタの内容をASCII文字とみなし、対応する2進数に変換する。但し、対応する2進数に変換されるのは、ASCII文字0, 1, ………9, A,………F に対してだけである。

```
    LD      A，'0'
    CALL        ASBIN
```

注　A，Fの内容は変化する。

9.　PRMES

指定チャンネルCHANELへ文字列を出力する。

```
MESSGE:  DEFM     'HELLO!'<3>
           ⌇
         LD       E,CHANEL
         LD       HL,MESSGE
         CALL     PRMES
```

注1.　A，F，Dは変化する。

注2.　HLは文字列の先頭アドレスを示す。文字列の最後にはETX（03H）をつける。

　　Eはチャンネル

10.　EXIT

モニタへ制御は復帰する。

```
JP       EXIT
*  ← モニタが打ち出してくる。
```

11.　FEEDER

指定チャンネルCHAN9へ256個のNULL（00H）を出力する。

```
LD       E,CHAN9
CALL     FEEDER
```

注　Eの内容は不変。

　A，F，Dの内容は変化する。

12.　GETARG

最大2個までの引数列を入力する。フォーマットはモニタ・コマンドと同じ。

```
LD       E,CHANEL
CALL     GETARG
```

サブルーチン・コール後の状態。

　B：引数の個数　0はターミネータのみ

　A：ターミネータ（CR，LF，；，／，↑のいずれか）に対応するASCIIコード

HL：第1引数の内容

ARGBUF：FF0B	第1引数の値	(下位)
0C	〃	(上位)
0D	第2引数の値	(下位)
0E	〃	(上位)

CHANELの値

CHANELの値は、モニタであらかじめ決められている。

CI	Console	Input	Channel	E=0
CO	Console	Output	Channel	E=0
OI	Object	Input	Channel	E=1
OO	Object	Output	Channel	E=1
SI	Source	Input	Channel	E=2
SO	Source	Output	Channel	E=2

SM-B-80T
ユーザーズマニュアル

7

目　　次

第1章　SM-B-80T概要

シャープSM-B-80Tは、これからマイクロコンピュータを理解し、実際に使ってみようという方、あるいは、実際にソフトウェアのデバッグに使用したり、ハードウェアに応用しようという方のために開発されたトレーニング用ワンボード・マイクロコンピュータです。

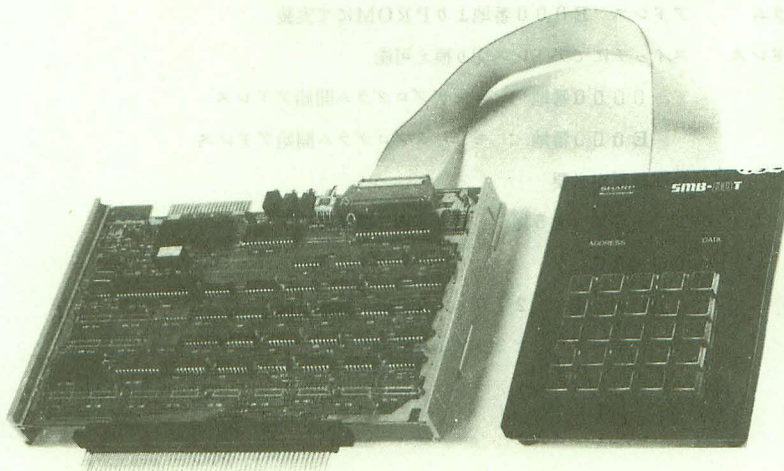
SM-B-80Tは次のような特長をそなえております。

(1)　SM-B-80Tは、完全組み立て済みボードであるため、電源を接続するだけでキースイッチよりプログラムをメモリに書き込んで、その実行ができます。

(2)　オーディオカセットテープレコーダとのインターフェースを内蔵しており、開発したプログラムをカセットテープに記録保存し、必要なときに自由に再生ができます。
（リモート端子により自動、または、マニュアルによるカセットのスタート／ストップが可能）

(3)　ボード上でメモリ（RAM，ROM）、パラレルI/Oポート、バスドライバを増設可能です。

(4)　キーボードとのインターフェースに使用しているPIOは、キーボードを使用しないときは単独使用が可能です。

(5)　システムに組み込み可能なボードサイズ（CPUボード）を採用し、外部との接続に100ピンコネクタを設けています。（270 × 190㎜　3.175㎜ピッチ）

1.1　SM－B－80Tの仕様

CPU	LH－0080（Z－80CPU）
クロック	2.4576MHz　（4.9152MHzクリスタル使用）
ROM	LH－7055×1個　1Kバイト実装（モニタプログラム）　MAX2Kバイト EPROMも可能（2708タイプ）
RAM	LH－2111A4×2個　256バイト実装 LH－2114－3×2個　1Kバイト実装 MAX　3.25Kバイト
シリアルI/Oポート	オーディオカセットテレコとの入出力専用 入力端子 ： イヤホン端子または外部スピーカ端子 出力端子 ： AUX端子またはLINE端子 制御端子 ： REM 転送速度 ： 300ビット／秒 変調方式 ： FM チェックサムを実施
パラレルI/Oポート	LH－0081×1個　8ビット×2ポート（キーボードインターフェース） MAX　LH－0081×2個
入力装置	キーボードスイッチ　　25個 データキー　　16個 ファンクションキー　　9個
出力装置	8桁7セグメントLEDによる16進数表示 レジスタ名についてはシンボル表示
動作モード	シングルステップ（1命令実行）＆オート実行
モニタプログラム	アドレス　E000番地よりPROMにて実装
リスタートアドレス	スイッチにてアドレス切り換え可能 0000番地 ： ユーザプログラム開始アドレス E000番地 ： モニタプログラム開始アドレス
電　源	外部電源が必要 ＋5V　±5％　　最大2A（標準構成）
動作温度	0～40℃
外形寸法	CPUボード ： 270×190mm　基板寸法 キーボード ： 139×190mm

第2章　取り扱い方法

SM-B-80Tの動作確認は図2.1に示す順序で行ってください。

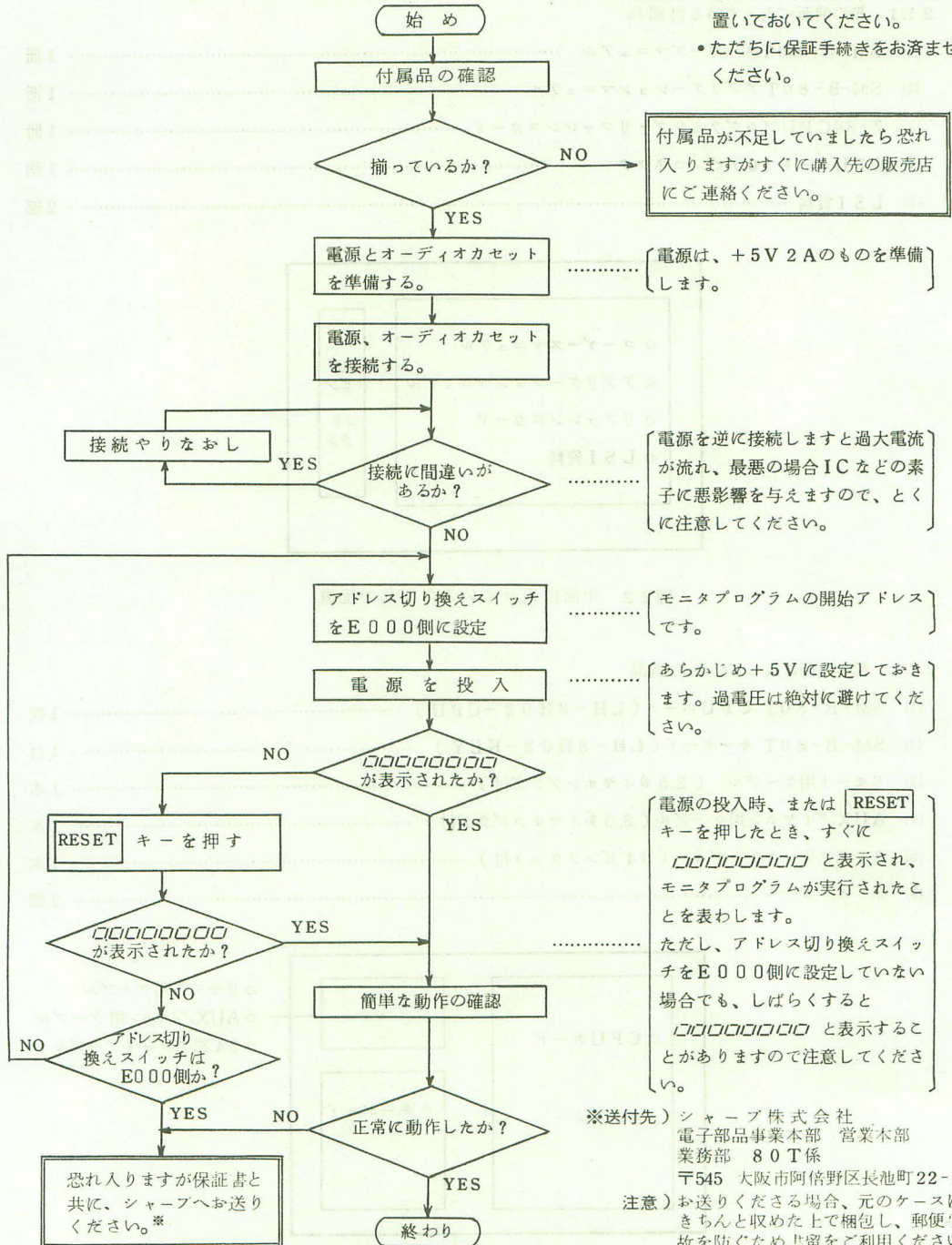



図2.1　SM-B-80Tの取り扱い方法

2.1　付属品の確認

SM-B-80Tには、次の付属品が含まれています。必ず確認してください。

2.1.1　箱の前面に入っている付属品

(1)　SM-B-80Tユーザーズマニュアル …………………… 1冊

(2)　SM-B-80Tアプリケーションマニュアル …………………… 1冊

(3)　Z-80CPUプログラミング・リファレンスカード …………………… 1冊

(4)　CPUボード100ピンコネクタ …………………… 1個

(5)　LSI資料 …………………… 2部

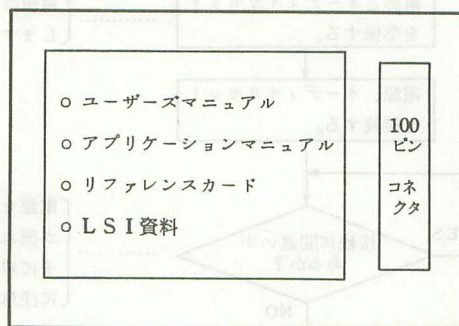



図2.2　中面に入っている付属品の配置

2.1.2　箱の内側に入っている付属品

(1)　SM-B-80T CPUボード(LH-8H02-CPU) …………………… 1枚

(2)　SM-B-80T キーボード(LH-8H02-KEY) …………………… 1台

(3)　リモート用ケーブル　(2.5φイヤホンプラグ付) …………………… 1本

(4)　AUX／イヤホン用ケーブル(3.5φイヤホンプラグ付) …………………… 1本

(5)　34芯フラットケーブル　(34ピンソケット付) …………………… 1本

(6)　ホ　ル　ダ　ー …………………… 2個

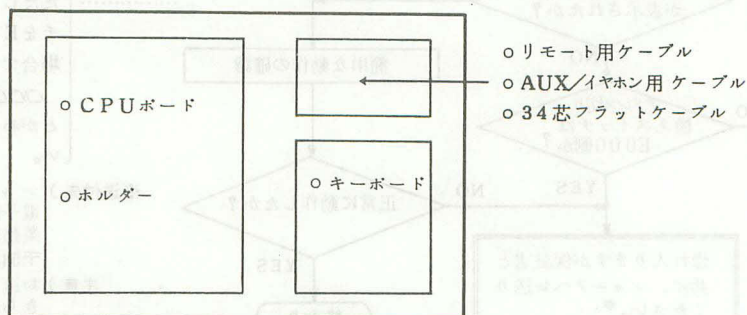



図2.3　内側に入っている付属品の配置

2.2　ホルダーの使い方

ＣＰＵボードの両端に取り付けられているホルダーは、プリント板を浮かすためのものです。プリント板を増すときにはホルダー同志を重ね合わせられるようになっています。

ホルダーでプリント板は約２cm浮いた状態になりますが、安全を考えてプリント板の下に導電性のあるもの（金属類など）は絶対に置かないでください。

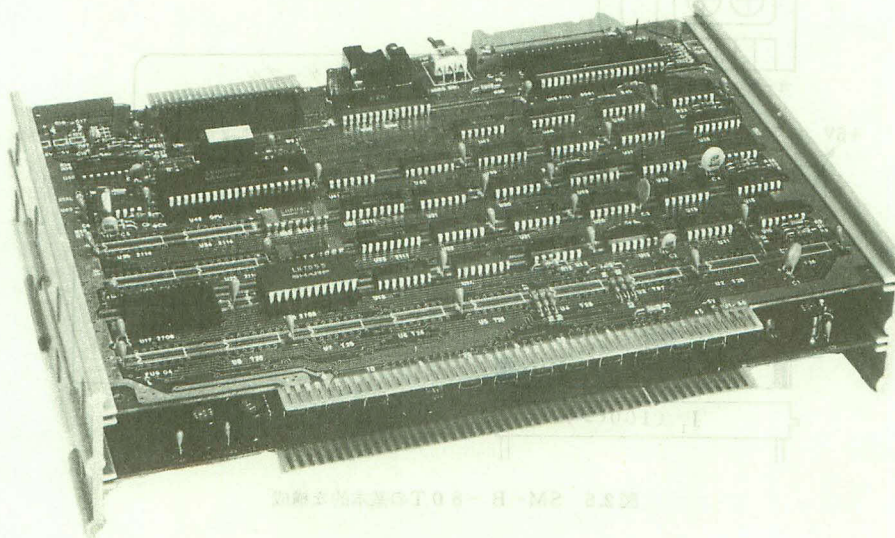



図2.4　ホルダーの使い方

2.3　道具の準備

SM-B-80 Tを動作させるに必要な道具は次のようになります。

(1)　直流安定化電源　　　　5Ｖ　最大２Ａ（標準構成）×１台

(2)　オーディオカセットテープレコーダ

ＡＵＸ端子またはＬＩＮＥ端子とイヤホン端子、または外部スピーカ端子がついているものが必要です。

リモート端子がついてなくても使用できますが、ついていれば自動スタート／ストップができますので便利です。

(注)　オーディオカセットのイヤホン端子がクリスタルイヤホン専用となっている場合には、インピーダンスの差により誤動作しますので、この用途には不適当です。

2.4 接続方法

SM-B-80Tの接続は基本的には次のようになります。また、キーボードとの接続はJ_2(34ピン)の1ピン表示と34芯フラットケーブルの1ピン表示(着色部)を合わせて接続します。

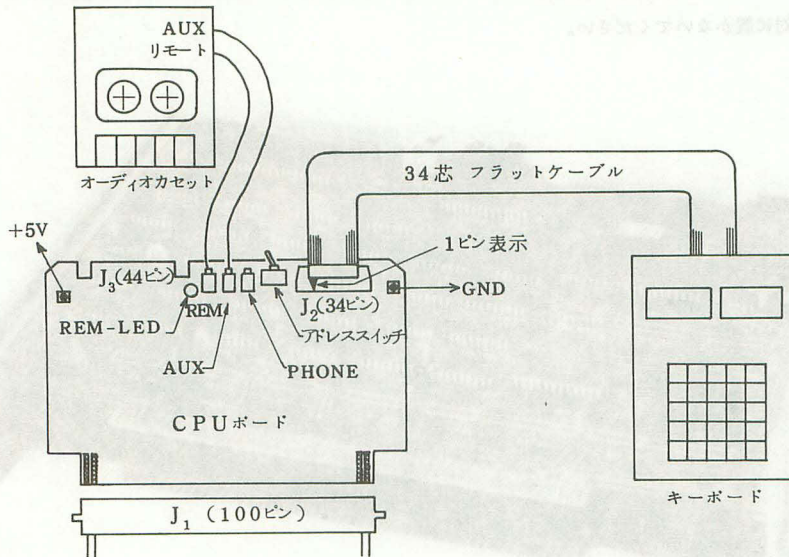



図2.5 SM-B-80Tの基本的な構成

2.4.1 電源の接続

電源を接続するために2本の電線が必要ですが、+5Vに赤色、0V(GND)に黒色の線を使うのがよいでしょう。

電源を接続するには、図2.6のようにプリント板の電源配線部分に半田付けする方法と、図2.7に示したJ_1(100ピンコネクタ)のピンに半田付けする方法があります。

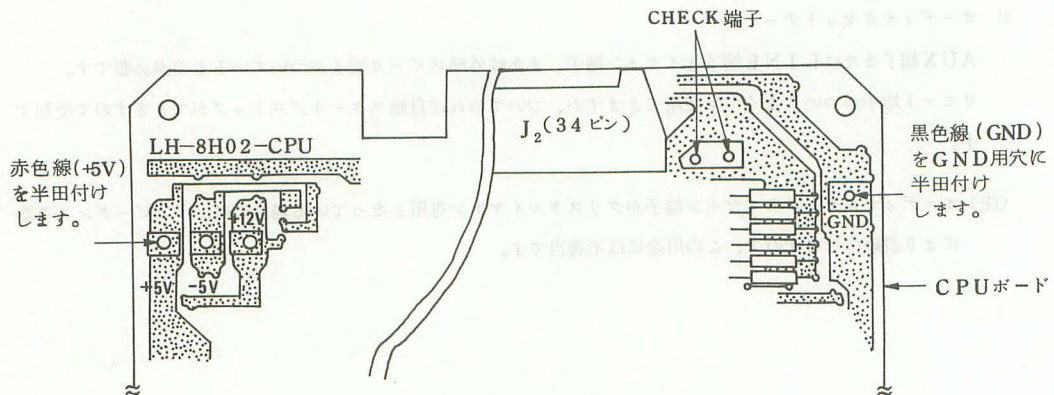



図2.6 プリント板へ半田付けの場合

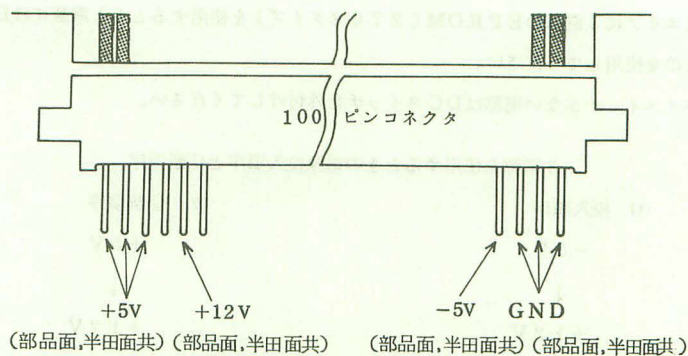



図2.7　J_1（100ピン コネクタ）へ半田付けの場合

＋12Ｖと－5Ｖの電源は、3電源のＥＰＲＯＭ（２７０８タイプ）を使用するときのためのものです。詳細は6.4項のＥＰＲＯＭ使用方法を参照してください。

2.4.2　電源に関する注意事項

電源には、ACスイッチとDCスタンバイスイッチの両方が付いているものを使用してください。

ACスイッチだけが付いている電源を使用するときは、図2.9のようにDCスイッチを付けてください。

(注) 電源のACスイッチ ON／OFF 時に電源トランスに発生するサージ電流で、DC出力にスパイク上の異常電圧が乗ることがあります。

最悪の場合にはICなどの素子に悪影響を与えます。

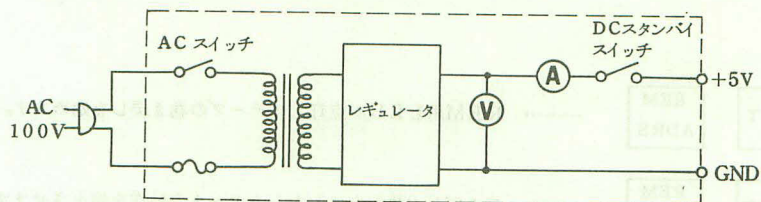



図2.8　好ましい電源

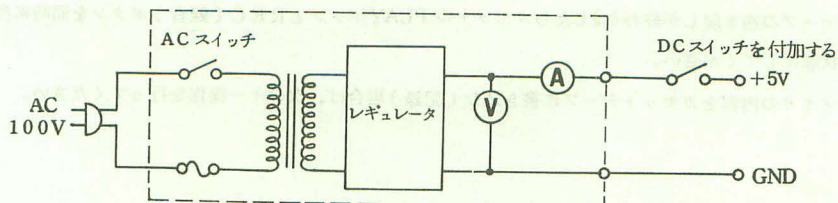



図2.9　DCスイッチを取り付けた場合

ユーザROMエリアに3電源のEPROM（2708タイプ）を使用するときも電源にはDCスタンバイスイッチが付いたものを使用してください。

DCスタンバイスイッチがない電源はDCスイッチを外付けしてください。

3電源を使用するときの電源投入順序と切断順序

2.4.3　オーディオカセットとの接続

メモリに入っているデータ（プログラム）をカセットテープに記録したり、カセットテープに記録したデータをメモリに読み込んだりできます。これにより作成したプログラムをカセットテープに記録して保存できます。また、必要なときにはいつでもカセットテープに記録したプログラムを再生できます。

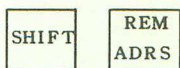
(1) メモリの内容をカセットテープに記録（カセットテープへのストア）

(a) オーディオカセットにカセットテープをセットします。

(b) リモート用ケーブルの一方をCPUボードのREM端子へ、他方をカセットのリモート端子へ差し込みます。

(c) AUX／イヤホン用ケーブルの一方をCPUボードのAUX端子へ、他方をカセットのAUX端子へ差し込みます。

(d) カセットの巻き戻しボタンを押してカセットを巻き戻し状態にしてから次のキー操作で巻き戻してください。

……… REM用LEDが点灯してテープの巻き戻しを始めます。

……… 巻き戻しが終わりましたらカセットの動作を停止させます。

(e) テープの巻き戻しが終わりましたらカセットのPLAYボタンとREC（録音）ボタンを同時に押して録音状態にしてください。

(f) メモリの内容をカセットテープに書き込む（記録）場合は、次のキー操作を行ってください。

(2) カセットテープよりの読み込み(カセットテープよりのロード)

(a) オーディオカセットにカセットテープをセットします。

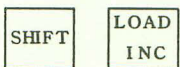
(b) リモート用ケーブルの一方をCPUボードのREM端子へ、他方をカセットのリモート端子へ差し込みます。

(c) AUX/イヤホン用ケーブルの一方をCPUボードのPHONE端子へ、他方をカセットのイヤホン端子へ差し込みます。

(d) カセットの巻き戻しボタンを押して巻き戻し状態にしてから次のキー操作でテープを巻き戻してください。

(e) テープの巻き戻しが終わりましたらPLAYボタンを押して再生状態にし、カセットの音量ツマミを中間〜最大の間にしてください。

また、音質調整ツマミがあればこれも中間〜最大の間にしてください。

(f) カセットテープの内容をメモリに読み込みたい場合は、次のキー操作を行ってください。

(注) カセットテープを巻き戻す場合は、磁気テープの部分で停止してください。
(テープの頭の部分には磁気テープでない所があります。)

2.5 システムの拡張

SM-B-80T は、ボード内で簡単にメモリ、パラレル I/Oポート、バスドライバを増設できますが、その方法について説明します。

2.5.1 メモリの増設

ボード内では、ROM1Kバイト、RAM1.25Kバイトを実装していますが、さらに、ROMを1Kバイト、RAMを2Kバイト増設できます。

増設方法を次に示します。

(1) ROMを増設するときは、図2.10 に示す部品番号 U_{17} の位置に取り付けます。

(2) RAMを増設するときは、1Kバイト単位に図2.10 に示す部品番号の位置に次のように取り付けます。

o 1Kバイト増設の場合 ……… U_{25} と U_{34} の位置に取り付けます。

o 2Kバイト増設の場合 ……… U_{25} と U_{34}, U_{26} と U_{35} の位置に取り付けます。

(注) ROMにEPROMを使用する場合は、6.4項(第6章)を参照してください。

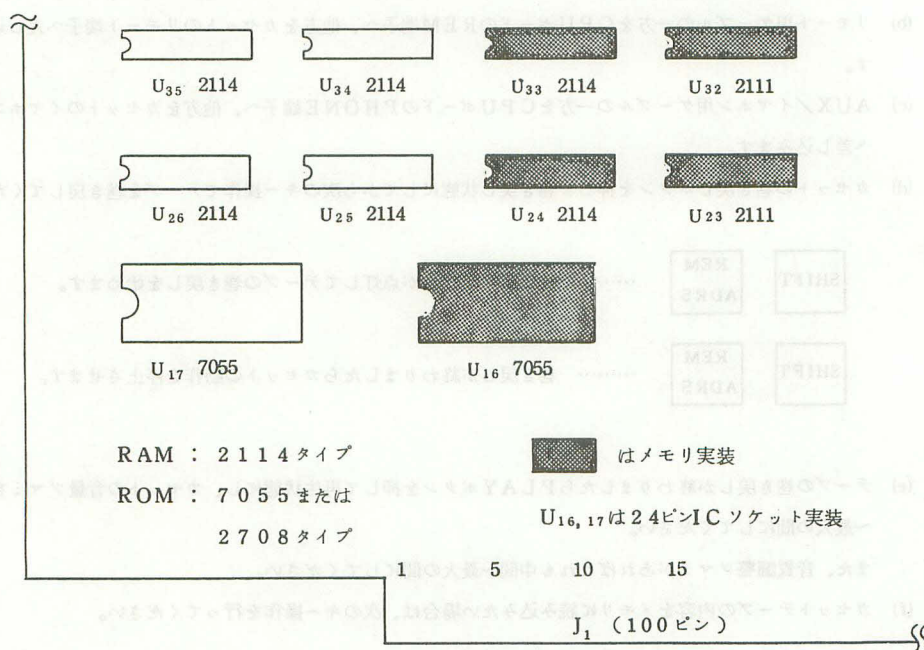



図2.10 メモリの増設

2.5.2 パラレル I/Oポート(PIO)の増設

パラレルの I/Oポートとして8ビット×2ポート(Z-80 PIO×1個)増設できます。

PIOを増設するときは、図2.11 に示す部品番号 U_{51} の位置に取り付けます。

また、このPIOの I/Oポートの信号ラインは J_3 (44ピンコネクタ端子)に配線されています。

J_3 の端子配列は付録5の端子配列図を参照してください。

J_3 適合コネクタ ： 4600－044－112　ケル㈱製

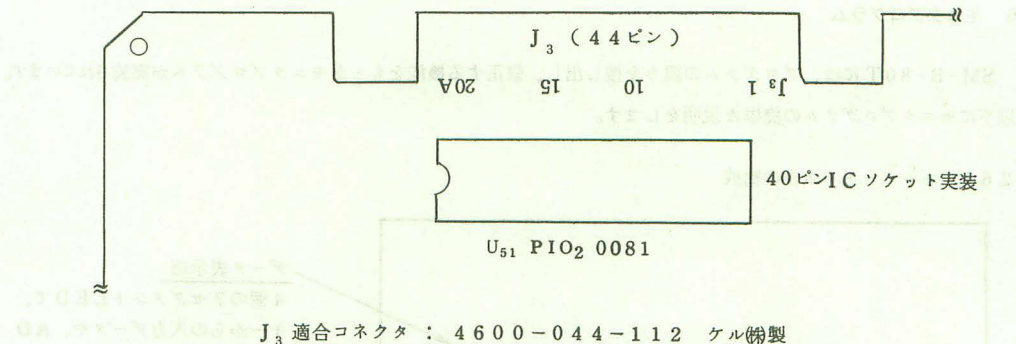



図2.11　P I Oの増設

2.5.3　バスドライバの増設

SM-B-80Tでは、双方向性アドレスバス、双方向性データバス、コントロール信号用バスを設けていますので、システムを拡張したいときは図2. 12に示す部品番号U_1 ～ U_9 の位置に取り付けてください。

J_1 （100ピンコネクタ端子）の各信号の端子配列は、付録5の端子配列図を参照してください。

ｏ双方向性アドレスバス　………　U_5 ～ U_8 の位置に8T26を取り付けます。

ｏ双方向性データバス　………　$U_{1,2}$ の位置に8T26を取り付けます。

ｏコントロール信号用バス　……　U_3 の位置に8T97を、U_4 の位置に8T28を、U_9 の位置に7404を取り付けます。

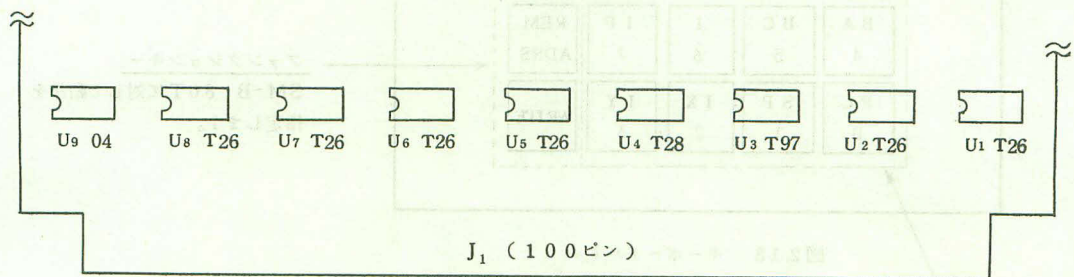



図2. 12　バスドライバの増設

双方向性データバスの制御方法については6.7項を参照してください。

（注）ICをプリント板に半田付けする際に使用する半田ゴテは、アース付きのものを使用してください。

2.6 モニタプログラム

SM-B-80Tには、プログラムの誤りを捜し出し、修正する機能をもったモニタプログラムが実装されています。以下にモニタプログラムの簡単な説明をします。

2.6.1 キーボードパネルの構成

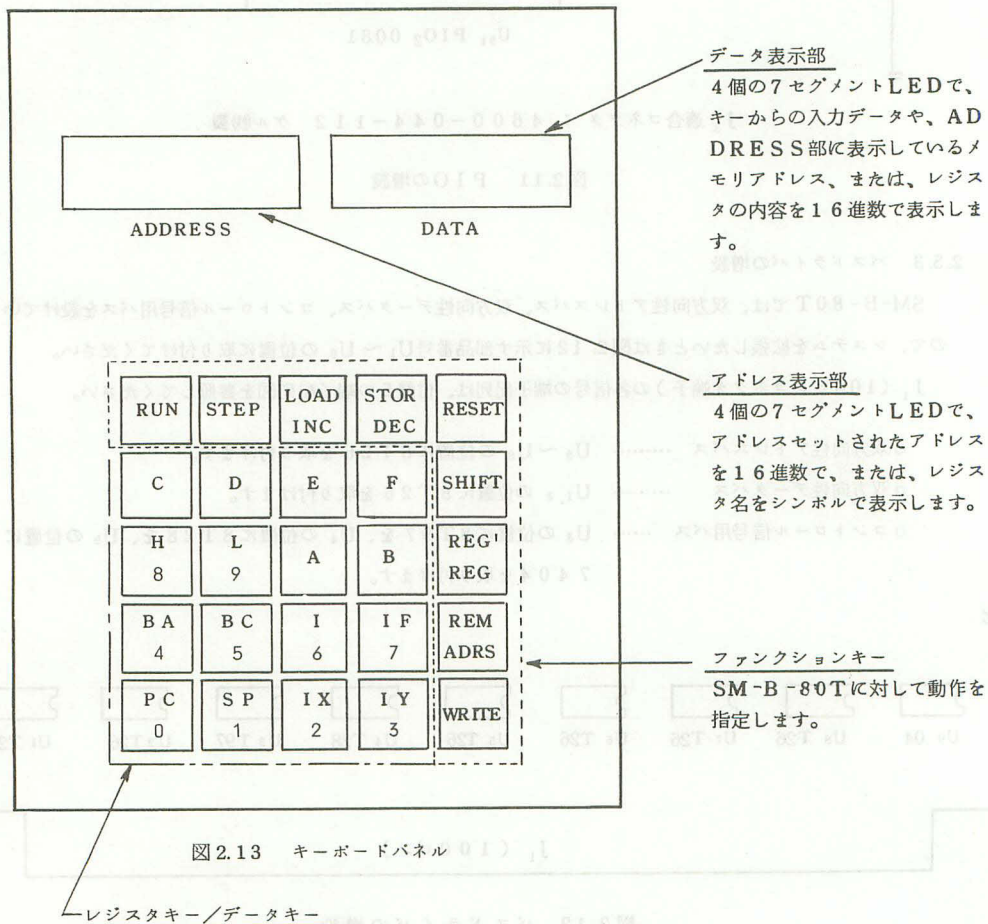



図2.13　キーボードパネル

○レジスタキー（上部分とA～Fキー）

　レジスタの指定に使用します。

○データキー　（下部分とA～Fキー）

　16進数を入力するときに使用します。

2.6.2 キーボードスイッチとコマンド

SM-B-80Tで使用するキーボードスイッチのコマンドについて説明します。

キー	説明
RESET	プログラム異常（プログラム暴走）によるCPU停止などに対して、システムを初期状態に戻します。
SHIFT	このキー操作後、ダブルファンクション構成キーの上部コマンドが有効になります（ファンクションキーの青色文字のコマンド）。
REG' / REG	REG'：レジスタの内容を表示させるときに、補助レジスタを指定します。 REG ：レジスタの内容を表示させるときに、主レジスタを指定します。
REM / ADRS	REM ：オーディオカセット用のリモートスイッチをON／OFFします。 ADRS：データ表示部に表示している16進数4桁のデータを、アドレスとしてアドレス表示部に表示し、そのアドレスのメモリ内容をデータ表示部に表示します。
WRITE	データ表示部の下位2桁に表示している16進数データを、アドレス表示部に表示しているメモリのアドレスへ書き込み、アドレス表示を＋1します。 あるいは、データ表示部の下2桁、または、4桁に表示している16進数データを、アドレス表示部に表示しているレジスタに書き込み、次のレジスタ名を表示します。
RUN	アドレス表示部に表示しているアドレスからユーザプログラムを実行します。
STEP	プログラムカウンタ（PC）が示しているアドレスからユーザプログラムを1命令実行します。
LOAD / INC	LOAD ：カセットテープに記録されたプログラム（16進データ）を、そのプログラムで指定されているアドレスのメモリへ書き込みます。（プログラムのロード） INC ：アドレス表示部に表示しているアドレスを＋1し、データ表示部にそのアドレスのメモリの内容を表示します。 または、アドレス表示部に表示しているレジスタ名を次のレジスタ名に変更し、データ表示部にそのレジスタの内容を表示します。
STOR / DEC	STOR ：アドレス表示部に表示しているアドレスから、データ表示部に表示しているアドレスまでのメモリの内容を、カセットテープに記録します。（プログラムのストア） DEC ：アドレス表示部に表示しているアドレスを－1し、データ表示部にそのアドレスのメモリの内容を表示します。 または、アドレス表示部に表示しているレジスタ名を前のレジスタ名に戻し、データ表示部にそのレジスタの内容を表示します。
データキー	0〜F ：データ（16進数）の入力に使用します。
レジスタキー	アドレス表示部にレジスタ名を表示し、データ表示部にそのレジスタの内容を表示します。 PC　プログラムカウンタ SP　スタックポインタ IX　インデックスレジスタX IY　インデックスレジスタY

BA	ブレークアドレスレジスタ	
BC	ブレークカウンタ	
I	インターラプトページアドレスレジスタ	
IF	インターラプトイネーブルフラグレジスタ	
A(A')	アキュムレータ	
F(F')	フラグレジスタ	
B(B')	Bレジスタ	
C(C')	Cレジスタ	( )内は補助レジスタ
D(D')	Dレジスタ	
E(E')	Eレジスタ	
H(H')	Hレジスタ	
L(L')	Lレジスタ	

2.6.3 表示

16進数とレジスタ名は、7セグメントLEDに次のように表示します。

(1) 16進数キーと表示

キー	表示	キー	表示	キー	表示	キー	表示
0	o	4	4	8	8	C	C
1	1	5	5	9	9	D	d
2	2	6	6	A	A	E	E
3	3	7	7	B	b	F	F

表 2.1 16進数の表示

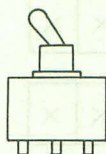
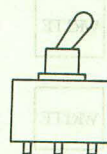
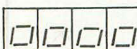
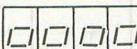
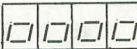
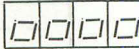
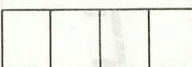
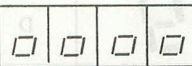
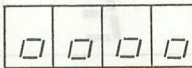
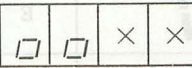
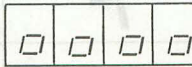
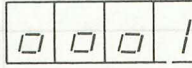
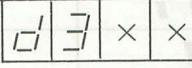
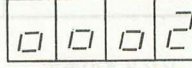
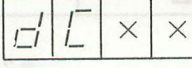
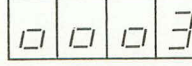
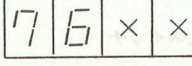
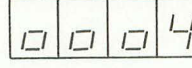
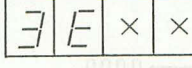
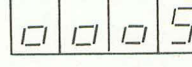
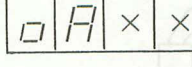
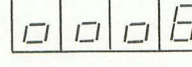
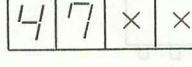
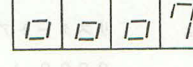
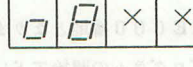
(2) レジスタキーと表示

レジスタを表示させる場合、主レジスタは [REG' / REG] キーを、補助レジスタは [SHIFT] キーを押した後、[REG' / REG] キーを押してから下記レジスタキーで表示させます。

主　レ　ジ　ス　タ				補　助　レ　ジ　ス　タ	
キー	表　示	キー	表　示	キー	表　示
P C	PC	A	A	A	A'
S P	SP	B	b	B	b'
I X	IH	C	C	C	C'
I Y	Iy	D	d	D	d'
I	I	E	E	E	E'
I F	IF	F	F	F	F'
H	h			H	h'
L	L			L	L'

表 2.2　レジスタの表示

2.6.4　アドレス切り換えスイッチ

SM−B−80Tは、アドレス切り換えスイッチの設定を変えることによりCPUのリスタートアドレスを次のように変更できます。

リスタートアドレスはE000番地となります。　リスタートアドレスは0000番地となります。

E000 ： モニタプログラムの開始アドレス　　0000 ： ユーザプログラムの開始アドレス

2.7 簡単な動作の確認

SM-B-80Tが正常に動作しているかどうかを次の順序で確認してください。操作方法の詳細は、第3章を参照してください。

2.7.1 メモリの内容表示と変更（メモリへの書き込み）、キーボード確認

次のキー操作で7セグメントLEDの表示内容を確認してください。このキー操作の前の電源投入で、LEDが 0000 0000 を表示しているか確認してください。0000 0000 を表示していないときは RESET キーを入力してください。

キー操作			ADDRESS	DATA	
アドレススイッチ E000側					E000：モニタプログラム開始アドレス
電源投入			0000	0000	モニタプログラムスタート
REM ADRS			0000	00××	××：0000番地内容
PC 0	……………	F	0000	CdEF	キー操作に対応した表示は表2.1を参照してください。
D	IY 3	WRITE	0001	d3××	0000番地内容を16進数のD3に変更（メモリへの書き込み）
D	C	WRITE	0002	dC××	
IF 7	I 6	WRITE	0003	76××	
IY 3	E	WRITE	0004	3E××	
PC 0	A	WRITE	0005	0A××	
BA 4	IF 7	WRITE	0006	47××	
PC 0	H 8	WRITE	0007	08××	

キー	キー		ADDRESS	DATA
IX 2	SP 1	WRITE	0008	21××
D	BA 4	WRITE	0009	d4××
F	F	WRITE	000A	FF××
IF 7	BA 4	WRITE	000b	74××
IX 2	D	WRITE	000C	2d××
SP 1	PC 0	WRITE	000d	10××
F	C	WRITE	000E	FC××
C	IY 3	WRITE	000F	C3××
IX 2	C	WRITE	0010	2C××
E	PC 0	WRITE	0011	E0××

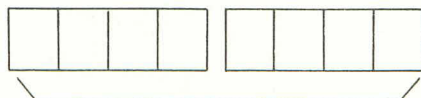
(注) ××は0〜Fの16進数字なら全て可です。

2.7.1でメモリに書き込んだデータ（プログラム）は、2.7.2〜2.7.4項の動作確認でも使用します。

2.7.2　0番地スタートの確認

ユーザRAM領域の開始アドレス、0000番地からプログラムを実行するかを確認します。

前項、2.7.1に引き続いて次のキー操作でREM端子用LEDが点灯することを確認してください。

キー操作	ADDRESS	DATA	REM・LED
アドレススイッチ 0000側			ユーザプログラム開始アドレス

消灯します。

2.7.3　ブレーク動作の確認

2.7.1項でメモリに書き込んだプログラムを使用してブレーク動作を確認します。2.7.2項に引き続いて次のキー操作で表示を確認してください。

キー操作	ADDRESS	DATA	
アドレススイッチ E000側			モニタプログラム開始アドレス
RESET	0000	0000	
REG'/REG　BA/4	bA-	××××	
PC/0　PC/0　PC/0　BC/5	bA-	0005	ブレークアドレスを0005番地に設定
WRITE	bC-	0000	
PC/0　SP/1　WRITE	1 -	00××	繰り返し回数を1回に設定
PC/0　PC/0　PC/0　IY/3	1 -	0003	2.7.1項のプログラムを0003番地より開始します。
REM ADRS	0003	033E	
RUN	0005	0A××	実行後、ブレークアドレス0005番地を表示して停止します。
	ブレークアドレス	アキュムレータ内容　フラグレジスタ内容	

（注）××は0～Fの16進数字なら全て可です。

プログラム実行をブレーク動作で行うと、ADDRESS表示部にブレークアドレスを、DATA表示部にアキュムレータの内容とフラグレジスタの内容を表示してプログラム実行を停止します。

2.7.4　ユーザレジスタの内容表示と変更、1命令実行、LED動作の確認

2.7.3項のブレーク動作確認に引き続き次のキー操作で表示を確認してください。LED動作は、D.P. も含めた全セグメントを点灯させて確認します。（D.P.はデシマル・ポイントの略）

キー操作	ADDRESS	DATA	
REG'/REG　A	A -	000A	0A：アキュムレータ内容
PC/0　H/8　WRITE	b -	00××	アキュムレータ内容を16進数の08に変更します。
STEP	0006	08××	0005番地からのプログラムを1命令実行して停止
REG'/REG　B	b -	0008	08：Bレジスタ内容
STEP	0007	××××	
SHIFT　REG'/REG　A	A' -	0008	08：補助アキュムレータ(A')内容
STEP	000A	××××	
RUN	8.8.8.8.	8.8.8.8.	LED全セグメント点灯

（注）××は0～Fの16進数なら全て可です。

1命令を実行してプログラム実行を停止するときは、ADDRESS表示部に次の命令の先頭アドレスを、DATA表示部に1命令実行後のアキュムレータとフラグレジスタの内容を表示します。

2.7.5　メモリ（1Kバイト）動作の確認

1KバイトRAM（LH－2114－3×2個）に1バイト単位で16進数の00、または、FFを書き込んだ直後に読み出しを行い、メモリが正常であることを確認します。

次のキー操作でプログラムをメモリへ書き込んでください。プログラムの格納には、モニタワーキングエリア用RAMを使用します。

キー操作				ADDRESS	DATA
F	F	PC 0	PC 0	××××	FF00
REM ADRS				FF00	00××
PC 0	I 6	WRITE		FF01	06××
F	F	WRITE		FF02	FF××
IX 2	SP 1	WRITE		FF03	21××
F	F	WRITE		FF04	FF××
PC 0	IY 3	WRITE		FF05	03××
IF 7	PC 0	WRITE		FF06	70××
IF 7	E	WRITE		FF07	7E××
B	H 8	WRITE		FF08	B8××
IX 2	PC 0	WRITE		FF09	20××
PC 0	B	WRITE		FF0A	0B××
IF 7	C	WRITE		FF0B	7C××

			ADDRESS	DATA
B	BC 5	WRITE	FF0C	b5 × ×
IX 2	B	WRITE	FF0d	2b × ×
IX 2	PC 0	WRITE	FF0E	20 × ×
F	I 6	WRITE	FF0F	F6 × ×
PC 0	BA 4	WRITE	FF10	04 × ×
IX 2	H 8	WRITE	FF11	28 × ×
F	PC 0	WRITE	FF12	F0 × ×
D	IY 3	WRITE	FF13	d3 × ×
D	C	WRITE	FF14	dC × ×
IF 7	I 6	WRITE	FF15	76 × ×
IX 2	IX 2	WRITE	FF16	22 × ×
E	SP 1	WRITE	FF17	E1 × ×
F	F	WRITE	FF18	FF × ×
BA 4	F	WRITE	FF19	4F × ×
E	D	WRITE	FF1A	Ed × ×
BA 4	IY 3	WRITE	FF1b	43 × ×
D	F	WRITE	FF1C	dF × ×

（注）××は0～Fの１６進数字なら全て可です。

メモリへのプログラム書き込みが終了し、書き込み誤りがなければ次のキー操作で、メモリの動作を確認します。

メモリが正常であればREM端子用LEDが点灯します。メモリに動作不良があると、最初の動作不良点のアドレスをADDRESS表示部に、DATA表示部上位２桁にメモリに書き込んだデータを、下位２桁にメモリから読み出したデータを表示してプログラムを終了します。

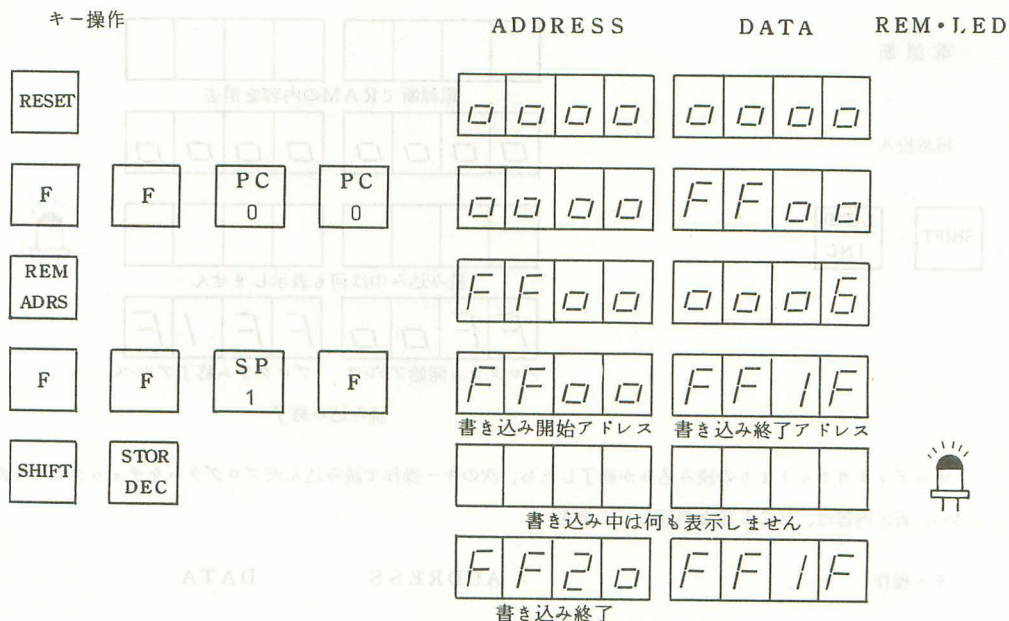
（動作不良例）

この例では、メモリアドレス０１００番地に書き込んだ１６進数のFFが、読み出しで７Fに変化しているのが確認できます。

2.7.6　オーディオカセットインターフェースの動作の確認

(1)　オーディオカセットへの書き込み

2.7.5項でメモリに書き込んだプログラムを次のキー操作でオーディオカセットに書き込んでください。

オーディオカセットに書き込む前には、オーディオカセットを録音状態にしてください。

オーディオカセットに書き込み中は、REM端子用LEDが点灯するだけで何も表示しません。

オーディオカセットへの書き込みが終了すると、ADDRESS表示部に書き込み終了アドレスの次のアドレスを、DATA表示部にはそのまま書き込み終了アドレスを表示します。

(注) REM端子をオーディオカセットに接続した状態でカセットテープの巻戻し、早送りを行いたいときは

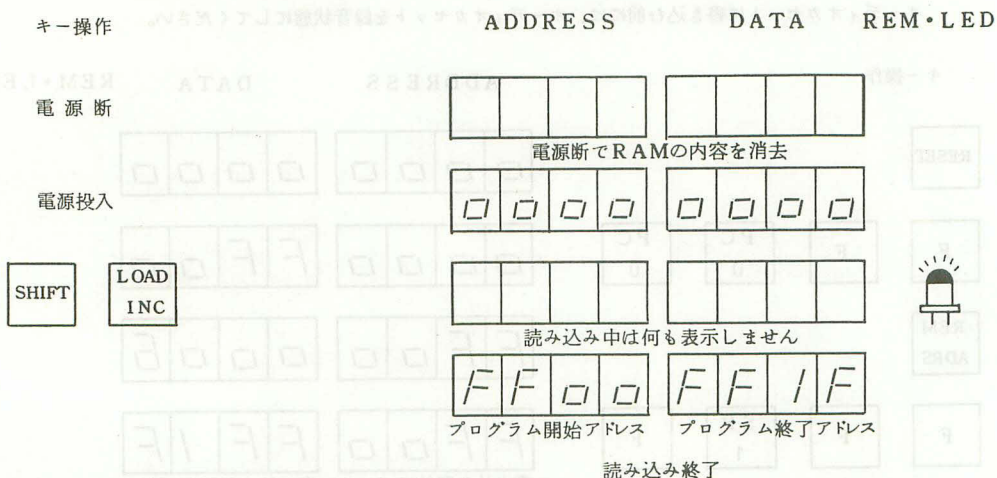
SHIFT　REM ADRS　キー操作でREM端子をONしてください。止めたいときは、同じキー操作をするか、

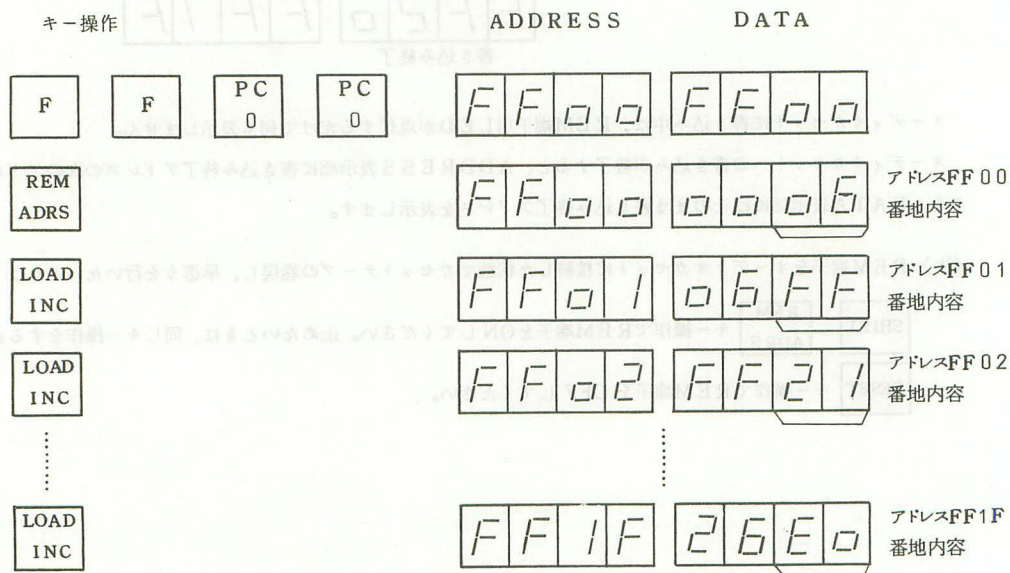
RESET　キー操作でREM端子を OFF してください。

(2) オーディオカセットよりの読み込み

(1)で書き込んだプログラムを、オーディオカセットより読み込みます。

プログラムを読み込む前に、プログラム書き込み前の位置にテープを巻き戻して再生状態にしてください。

キー操作	ADDRESS	DATA	REM・LED
電源断			
	電源断でRAMの内容を消去		
電源投入	0000	0000	
SHIFT　LOAD INC			(点灯)
	読み込み中は何も表示しません		
	FF00	FF1F	
	プログラム開始アドレス	プログラム終了アドレス	
	読み込み終了		

オーディオカセットよりの読み込みが終了したら、次のキー操作で読み込んだプログラムをチェックしてください。表示内容は、2.7.5項を参照してください。

キー操作	ADDRESS	DATA	
F　F　PC 0　PC 0	FF00	FF00	
REM ADRS	FF00	0006	アドレスFF00番地内容
LOAD INC	FF01	06FF	アドレスFF01番地内容
LOAD INC	FF02	FF21	アドレスFF02番地内容
⋮		⋮	
LOAD INC	FF1F	26E0	アドレスFF1F番地内容

第3章　操　作　説　明

この章では、SM-B-80Tを操作する上で必要なキーの使い方、表示の見方、オーディオカセットの使い方などを説明します。

3.1　システムのリセット

電源を投入したとき、または、RESET キーを押したときにシステムは初期状態に戻されます。また、初期状態に戻されたあとのリスタートアドレスは、2.6.4項で説明したようにモニタプログラム開始アドレスE000番地か、ユーザプログラム開始アドレス0000番地に変更できます。

3.2　モニタプログラムの基本的な操作方法

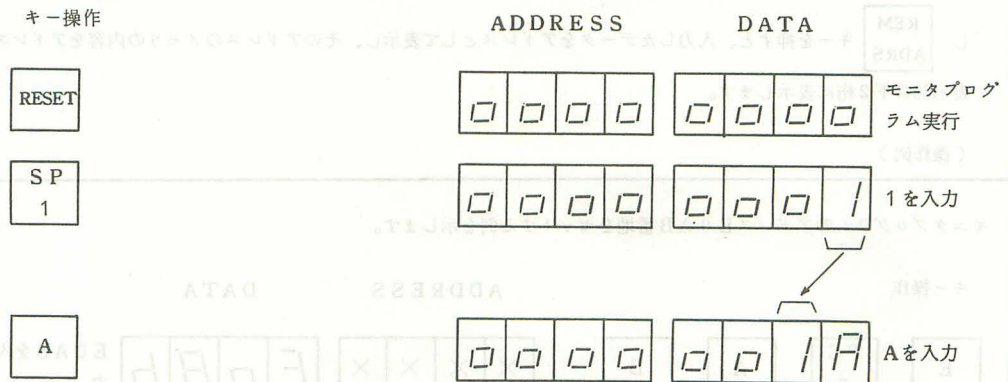
SM-B-80Tに実装していますモニタプログラムの操作方法を以下に説明します。

3.2.1　データのセット

セットしたいデータを16進数キー PC/0 ～ F で入力します。このとき押したキーに対応した文字をデータ表示部に表示します。

（操作例1）

16進数の1Aをセットする例を示します。アドレス切り換えスイッチはE000側に設定しておきます。
（3.2.1項以降もE000側に設定）

キー操作	ADDRESS	DATA	
RESET	0000	0000	モニタプログラム実行
SP/1	0000	0001	1を入力
A	0000	001A	Aを入力

（注） RESET キー操作後、表示は全て0になります。

（操作例２）

１６進数の１Ａを１Ｂと間違えて入力したときの修正方法を示します。

キー操作	ADDRESS	DATA	
RESET	0000	0000	モニタプログラム実行
SP/1　B	0000	001b	１Ａを１Ｂと間違えて入力
SP/1	0000	01b1	
A	0000	1b1A	１Ｂを１Ａに修正

間違えたデータ　正しいデータ

（注）メモリへ書き込むときは、データ表示部の下２桁のデータが有効です。

3.2.2　アドレスのセット

メモリへのデータ書き込み、メモリの内容表示やプログラムの実行をさせたいアドレスを１６進数でキー入力し REM/ADRS キーを押すと、入力したデータをアドレスとして表示し、そのアドレスのメモリの内容をアドレス表示部の下２桁に表示します。

（操作例）

モニタプログラムのアドレスＥ０ＡＢ番地をセットする例を示します。

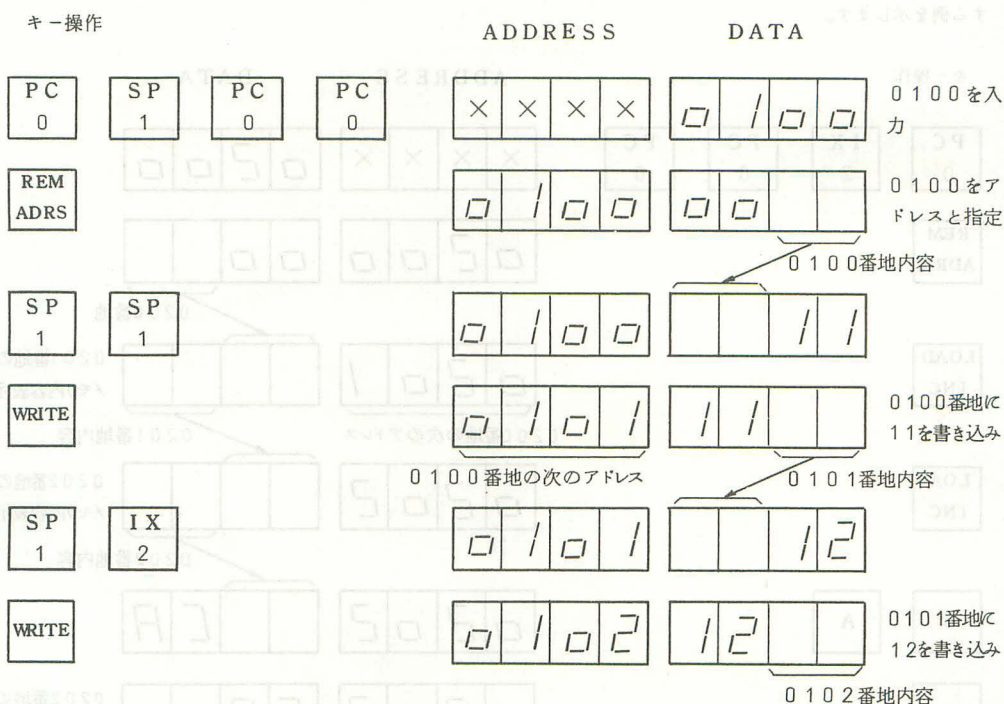
キー操作	ADDRESS	DATA	
E　PC/0　A　B	××××	E0Ab	Ｅ０ＡＢを入力
REM/ADRS	F0Ab	AbE1	Ｅ０ＡＢをアドレスと指定

Ｅ０ＡＢ番地内容

（注）××××は、Ｅ０ＡＢを入力する前の表示を表わします。

3.2.3　メモリへの書き込み

データを書き込みたいメモリのアドレスを１６進数で４桁キー入力し、 [REM ADRS] キーでアドレスとして指定します。次に書き込みたいデータを１６進数で２桁キー入力し、 [WRITE] キーを押すとセットしたアドレスのメモリにデータが書き込まれます。また、アドレス表示部は次のアドレスを示し、そのメモリの内容をデータ表示部下２桁に表示します。

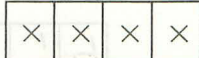
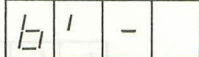
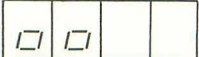
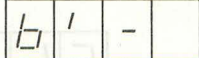
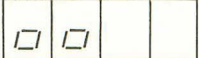
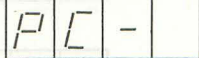
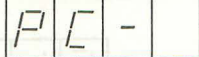
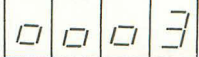
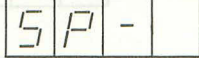
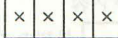
（操作例）

3.2.4　メモリの内容表示と変更

メモリの内容を表示させたいアドレスを１６進数でキー入力し、 [REM ADRS] キーでアドレスとして指定すると、そのアドレスのメモリの内容をデータ表示部下２桁に表示します。また、表示させたメモリの内容を変更したいときは、新しいデータをキー入力し、 [WRITE] キーを押すとメモリの内容が変更され、次のアドレスをアドレス表示部に、そのメモリの内容をデータ表示部に表示します。さらに次のアドレス、または、一つ前のアドレス

のメモリ内容を表示させたいときは [LOAD INC] キー、または、[STOR DEC] キーを押します。すると [LOAD INC] キーでは次のアドレスをアドレス表示部に、そのメモリの内容をデータ表示部に表示します。[STOR DEC] キーでは一つ前のアドレスをアドレス表示部に、そのメモリの内容をデータ表示部に表示します。

(注) [LOAD INC] キーはアドレスの表示を＋１します（次のアドレスを表示します）。
[STOR DEC] キーはアドレスの表示を－１します（一つ前のアドレスを表示します）。

（操作例）

０２００番地から０２０２番地のメモリの内容を表示させ、０２０２番地のメモリの内容を１６進数のＣＡに変更する例を示します。

(注) [× × × ×] は、０２００ を入力する前の表示を表わします。

3.2.5　ユーザレジスタの内容表示と変更

ユーザレジスタの内容を表示させたいときは [REG'/REG] キーを押した後レジスタ名に対応したキーを入力します。すると、そのレジスタのシンボル名をアドレス表示部に、レジスタの内容をデータ表示部に表示します。ただし、補助レジスタ（A' B' C' D' E' F' H' L'）の内容を表示させたいときは [REG'/REG] キーを押す前に [SHIFT] キーを押してください。また、表示させたレジスタの内容を変更したいときは、新しいデータをキー入力し、[WRITE] キーを押すとレジスタの内容が変更され、次のユーザレジスタのシンボル名をアドレス表示部に、そのレジスタの内容をデータ表示部に表示します。

（操作例１）

補助Bレジスタ（B'）とプログラムカウンタ（PC）の内容を表示させ、プログラムカウンタの内容を１６進数０００３に変更する例を示します。

キー操作	ADDRESS	DATA	
[SHIFT] [REG'/REG]	× × × ×	× × × ×	補助レジスタを指定
[B]	b' - （B' レジスタシンボル表示）	0 0 （(注)　B' レジスタ内容）	B' レジスタの内容を表示
[REG'/REG]	b' -	0 0	主レジスタを指定
[PC/0]	P C - （プログラムカウンタシンボル表示）	（プログラムカウンタの内容）	プログラムカウンタの内容を表示
[PC/0] [PC/0] [PC/0] [IY/3]	P C -	0 0 0 3	プログラムカウンタ内容を０００３に変更
[WRITE]	S P - （スタックポインタシンボル表示）	（スタックポインタ内容）	

(注) [×|×|×|×] は、[SHIFT] キーを押す前の表示を表わします。

ユーザレジスタには、８ビットのものと１６ビットの２種類があり、その表示方法にも違いがありますので表3.1を参照してください。

このほかにユーザレジスタの内容を連続して表示させたいときは、操作例1の方法でユーザレジスタの中のどれかを指定し、[LOAD/INC] キー、または、[STOR/DEC] キーで次々とレジスタの内容を表示させることができます。

（操作例2）

プログラムカウンタを指定し、レジスタの内容を連続して表示させる例を示します。

（注）ユーザレジスタの内容を表示させる順番は表3.1を参照してください。

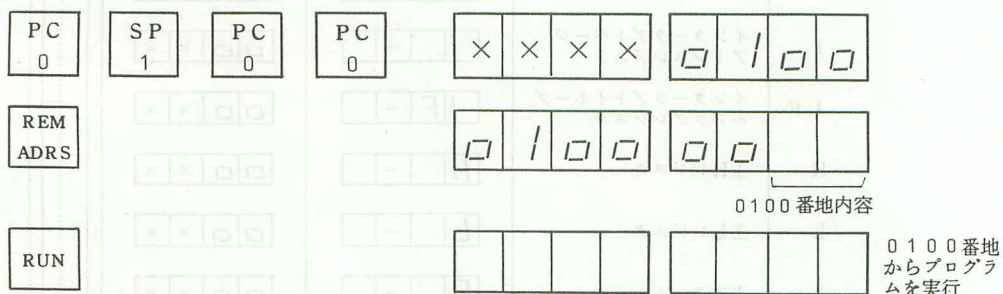
キー操作	ADDRESS	DATA	
[REG'/REG] [PC/0]	PC -	(プログラムカウンタ内容)	プログラムカウンタの内容を表示
[LOAD/INC]	SP - (スタックポインタシンボル表示)	(スタックポインタ内容)	スタックポインタの内容を表示
⋮			
[LOAD/INC]	F -	00 (Fレジスタ内容)	主レジスタFの内容を表示
[LOAD/INC]	h' -	00 (H'レジスタ内容)	補助レジスタH'の内容を表示
[LOAD/INC]	L' -	00 (L'レジスタ内容)	補助レジスタL'の内容を表示
⋮			
[LOAD/INC]	F' -	00 (F'レジスタ内容)	補助レジスタF'の内容を表示
[LOAD/INC]	PC -	(プログラムカウンタ内容)	一巡して最初のプログラムカウンタへもどり、その内容を表示
[STOR/DEC]	F' -	00 (F'レジスタ内容)	プログラムカウンタの一つ手前のレジスタF'の内容を表示
⋮			
[STOR/DEC]	F -	00 (Fレジスタ内容)	主レジスタFの内容を表示
⋮			
[STOR/DEC]	PC -	(プログラムカウンタ内容)	最初のプログラムカウンタへもどりその内容を表示

3.2.6 プログラムの実行

作成したプログラムをSM-B-80Tに実行させるには、プログラムの開始アドレスをアドレスセットし、RUN キーを押しますと、アドレスセットしたメモリアドレスよりプログラムを実行します。また、プログラムの最後にHALT命令を書き込んでおきますとHALT命令を実行した後、プログラムの実行を停止します。

（操作例）

アドレス0100番地からプログラムを実行させる例を示します。

キー操作	ADDRESS	DATA	
PC 0 / SP 1 / PC 0 / PC 0	××××	0100	
REM ADRS	0100	00	0100番地内容
RUN			0100番地からプログラムを実行

（注）×××× は、0100を入力する前の表示を表わします。

また、プログラムを0000番地から実行させる場合は、アドレススイッチを0000側に設定して RESET キーを押すことにより実行を開始できます。

3.2.7 ステップ動作

作成したプログラムに誤りが有る場合などに、そのプログラムを1命令ずつ実行させながら（ステップ動作）メモリやレジスタの内容を確認していけばプログラムの誤りの箇所を発見することができます。

ステップ動作をSM-B-80Tに行わせるには、ステップ動作をさせたいプログラムの開始アドレスを16進数でプログラムカウンタ（PC）に書き込み、STEP キーを押すとプログラムを1命令実行してプログラム実行を停止します。このため STEP キーを押して1命令実行して停止するごとにメモリの内容やレジスタの内容を確認することができ、プログラムのどの部分に誤りがあるかを調べることができます。

プログラムを1命令実行して停止したときにはアドレス表示部に次に実行するアドレスを、データ表示部の上位2桁にアキュムレータの内容を、下2桁にフラグレジスタの内容を表示します。

（操作例）

アドレス0100番地からプログラムをステップ動作させる例を示します。

ステップ動作でプログラム実行を停止した後、 RUN キーを押すとアドレス表示部に表示しているアドレスからプログラムを実行します。

3.2.8　ブレーク動作

ブレーク動作は、ブレークアドレスレジスタに書き込んだアドレスの一つ前までの命令をブレークカウンタに書き込んだ回数だけ実行して停止します。

ブレーク動作を使用しますと、プログラムの中に同じ命令をある条件になるまで何度も繰り返す命令があるときに、ステップ動作と組み合わせて使用すれば効率よくプログラムの誤りを発見できます。

ブレーク動作をSM-B-80Tに行わせるには、プログラム実行を停止させたいアドレス（ブレークポイント）をブレークアドレスレジスタ（ＢＡ）に書き込み、ブレークカウンタ（ＢＣ）には実行回数を書き込みます。そして、ブレーク動作をさせたいプログラムの開始アドレスをアドレスセットし、 RUN キーを押すと、ブレークアドレスレジスタ（ＢＡ）が指すアドレス（ブレークポイント）の一つ前の命令までのプログラムを、ブレークカウンタ（ＢＣ）が0になるまで実行して停止します。また、このときはアドレス表示部にブレークアドレスレジスタの内容（ブレークポイント）を表示し、データ表示部の上位２桁にアキュムレータ内容を、下２桁にはフラグレジスタ内容を表示します。

（注）ブレーク動作での繰り返し回数は、１～２５５回です。ブレークカウンタが0のときはブレーク動作は行わずに普通のプログラム実行となります。

ブレークポイントは、ＣＰＵの命令のオペレーションコード（ＯＰコード）が入っているアドレスでなければなりません。

オペレーションコードが入っているアドレス以外をブレークポイントとして設定しますと、ブレーク動作を行いません。

（操作例）

ブレーク動作をさせたいプログラムの開始アドレスを０１００番地とし、ブレークポイントを０１ＡＢ番地、繰り返し回数を１００回と設定してブレーク動作をさせる例を示します。

キー操作	ADDRESS	DATA	
REG'/REG　BA/4	bA-		ブレークアドレスレジスタを指定
		ブレークアドレスレジスタ内容	
PC/0　SP/1　A　B	bA-	01Ab	
WRITE	bC-	00	ブレークポイントとして０１ＡＢ番地を書き込み
		ブレークカウンタ内容	
I/6　BA/4	bC-	0064	繰り返し回数１００回を１６進数の６４で書き込み
WRITE	I-	00	
		Iレジスタ内容	
PC/0　SP/1　PC/0　PC/0	I-	0100	
REM ADRS	0100	00	０１００をアドレスとして指定
		0100番地内容	
RUN	01Ab		ブレークポイントでプログラム実行停止
	ブレークポイントを表示	アキュムレータ内容　フラグレジスタ内容	

（注）繰り返し回数は１６進数でブレークカウンタに書き込んでください。

繰り返し回数１００回＝１６進数の６４

ブレーク動作でプログラム実行を停止した後、 STEP キーを押すとブレークポイントよりステップ動作を実行します。また、 RUN キーを押したときはブレークポイントよりプログラムを実行します。

3.3 オーディオカセットとの接続

プログラムを入れるRAMメモリは、電源が切れるとその内容が壊われてしまい作成したプログラムの保存ができません。

そこでSM-B-80Tは一般のオーディオカセットをプログラム保存に使用しています。SM-B-80Tにはこのオーディオカセットとの間でプログラム（データ）をやりとりするためのインターフェースを内蔵しています。

3.3.1 プログラムのカセットテープへのストア（プログラムの書き込み）

プログラムをカセットテープへストアするときは図3.1に示すように接続してください。

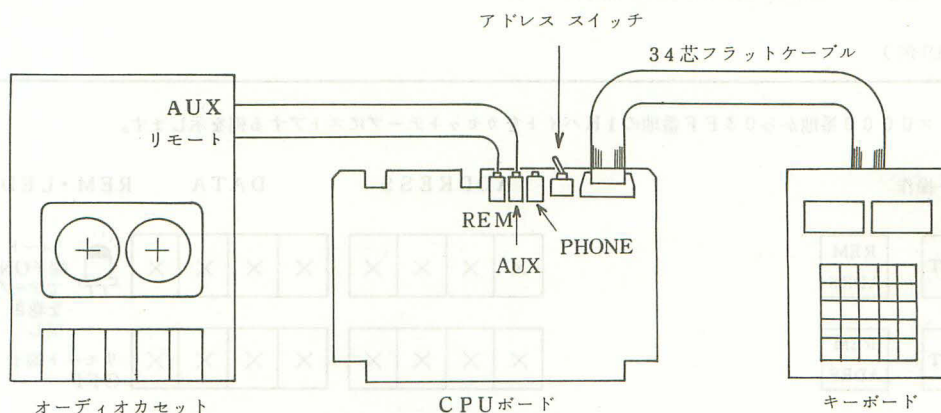



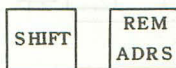
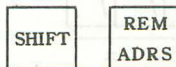
図3.1 プログラムをストアするときの接続

プログラムをカセットテープへストアするときは、次の手順で行ってください。

(1) オーディオカセットにカセットテープをセットします。

(2) リモート用ケーブルの一方をCPUボードのREM端子（EJ_1）へ、他方をカセットのリモート端子へ差し込みます。

(3) AUX／イヤホン用ケーブルの一方をCPUボードのAUX端子（EJ_2）へ、他方をカセットのAUX端子へ差し込みます。

(4) カセットの巻き戻しボタンを押して巻き戻し状態にしてから次のキー操作でテープを巻き戻してください。

リモート端子をONしてテープを巻き戻します。
このときREM用LEDが点灯します。

テープを巻き戻したらリモート端子を OFF します。

（注）リモート端子をキーで制御する場合は反転式になっていますので、リモート端子がONのときの [SHIFT] [REM ADRS] キー操作では OFF 、また、逆にリモート端子が OFF のときの [SHIFT] [REM ADRS] キー操作ではON状態になります。

(5) テープの巻き戻しが終わったらカセットのREC(録音)ボタンとPLAYボタンを同時に押して録音状態にしてください。

(6) 次にアドレス表示部に、ストアしたいプログラムが入っているメモリの開始アドレスをセットし、データ表示部にプログラムの終了アドレスをセットして [SHIFT] 次に [STOR DEC] キーを押すと自動的にカセットが動き出し、カセットテープへのストアを始めます。

ストアが終了すると自動的にカセットを停止して、データ表示部にストア前にセットしたプログラムの終了アドレスを、アドレス表示部に終了アドレスの次のアドレス(終了アドレス+1番地)を表示します。

また、ストア中はREM用LEDが点灯します。

(操作例)

3.3.2　プログラムのカセットテープよりのロード（プログラムの読み込み）

プログラムをカセットテープよりロードするときは図3.2に示すように接続してください。

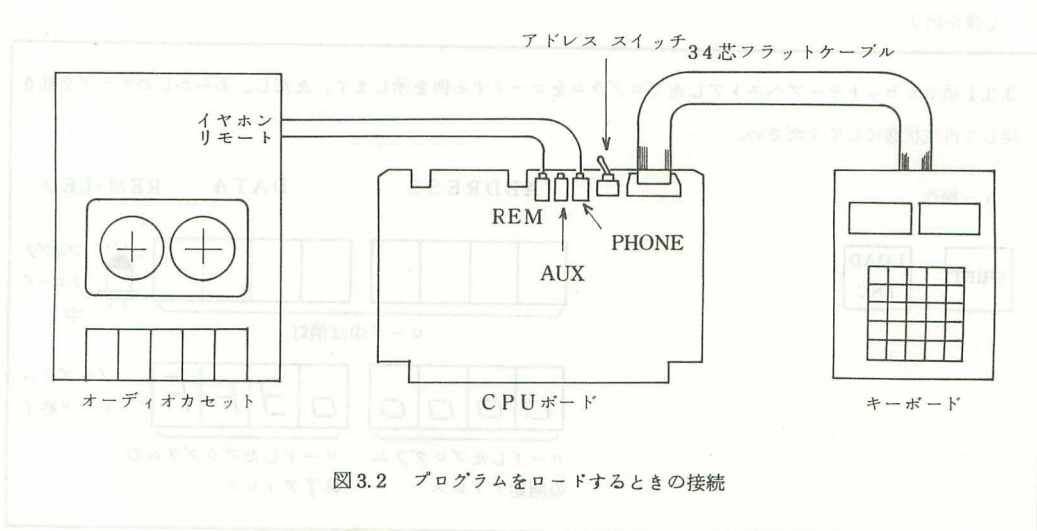



図3.2　プログラムをロードするときの接続

カセットテープからプログラムをロードするときは次の手順で行ってください。

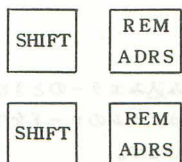
(1)　オーディオカセットにカセットテープをセットします。

(2)　リモート用ケーブルの一方をCPUボードのREM端子（EJ_1）へ、他方をカセットのリモート端子へ差し込みます。

(3)　AUX／イヤホン用ケーブルの一方をCPUボードのPHONE端子へ、他方をカセットのイヤホン端子、または、外部スピーカ端子へ差し込みます。

(4)　カセットの巻き戻しボタンを押して巻き戻し状態にしてから次のキー操作でテープを巻き戻してください。

リモート端子をONしてテープを巻き戻します。
このときREM用LEDが点灯します。

テープを巻き戻したらリモート端子をOFFします。

(5)　テープの巻き戻しが終りましたらカセットのPLAYボタンを押して再生状態にし、カセットテレコの音量ツマミを中間～最大の間にしてください。また、音質調整ツマミがあればこれも中間～最大の間にしてください。

(6)　次に SHIFT LOAD INC キーを押すと自動的にカセットが動き出しロードを始めます。

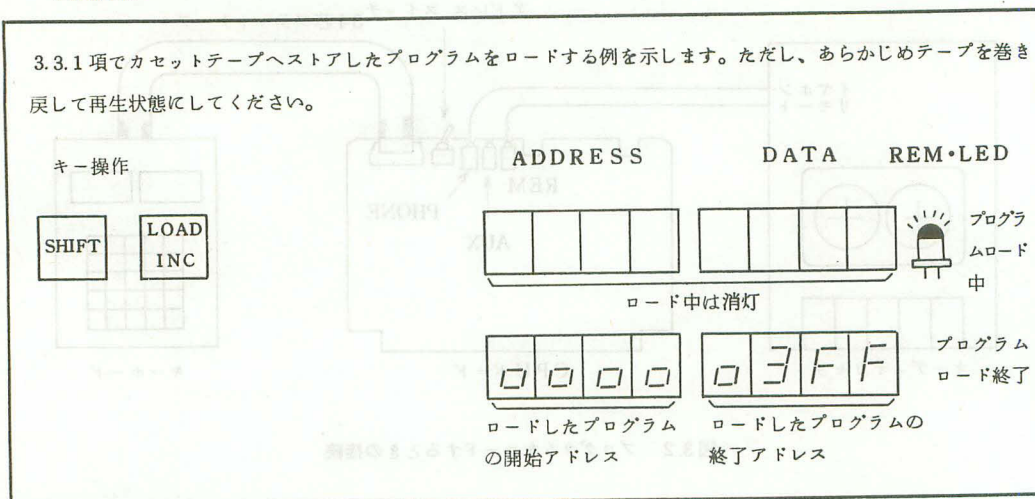
ロードが終了すると自動的にカセットを停止して、アドレス表示部にロードしたプログラムの開始アドレスを、データ表示部に終了アドレスを表示します。また、ロード中はREM用LEDが点灯します。

（注）カセットテープにストアするとき、最初の約30秒間はデータ1が書き込まれていますので、テープを巻き戻して再生状態にしたときは、この部分でテープを止めてください。

詳細は4.3.5項のカセットテープへの書き込みフォーマットの説明を参照してください。

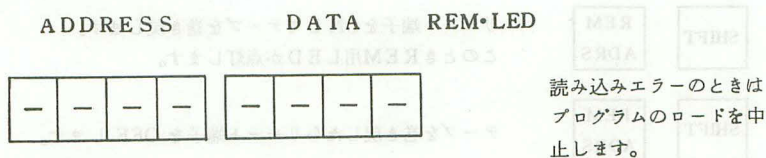
データ1の音をスピーカに再生すると、ピー（2.4 kHz）という音がします。また、データ0はポー（1.2 kHz）という音がします。

（操作例）

3.3.1項でカセットテープへストアしたプログラムをロードする例を示します。ただし、あらかじめテープを巻き戻して再生状態にしてください。

プログラムをロード中に次のような表示になったときは、プログラムの読み込みエラーですので最初からやりなおしてください。

（注）使用するカセットテープに傷がありますとプログラムの読み込みエラーの原因となりますので新しいものと交換してください。

プログラム読み込みエラーのときの表示

読み込みエラーのときはプログラムのロードを中止します。

3.4 プログラムの作成とデバッグの仕方

SM－B－80Tの基本的な操作方法がわかりましたので次に実際にプログラムを作成して、そのプログラムのデバッグの仕方を説明します。

3.4.1 プログラムの作成手順

SM－B－80Tを使用してのプログラムの作成手順を説明します。

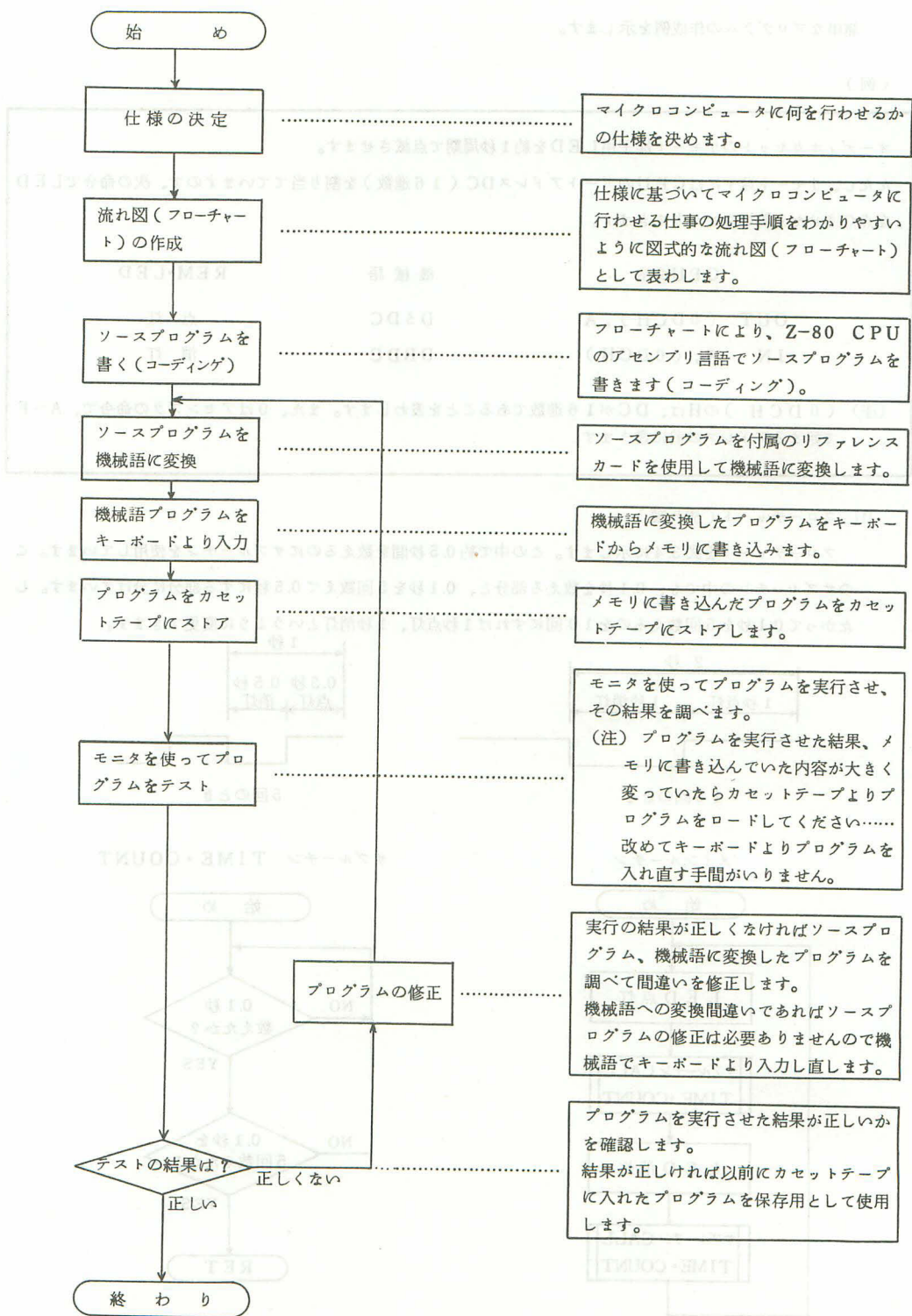



図3.3 プログラムの作成手順

簡単なプログラムの作成例を示します。

（例）

オーディオカセットのリモート端子用ＬＥＤを約１秒周期で点滅させます。

ただし、リモート端子にはＣＰＵのポートアドレスDC（１６進数）を割り当てていますので、次の命令でＬＥＤを点灯させたり消灯させたりできます。

ＣＰＵ命令	機械語	REM・ＬＥＤ
OUT （０DCH），A	D３DC	点 灯
IN A，（０DCH）	DBDC	消 灯

（注）（０DCH ）のHは、DCが１６進数であることを表わします。また、０はアセンブラの命令で、A～Fで始まるアドレスの前に書きます。

(1) フローチャート（流れ図）

フローチャートを図3.4に示します。この中で約0.5秒間を数えるのにサブルーチンを使用しています。このサブルーチンの中でも、0.1秒を数える部分と、0.1秒を５回数えて0.5秒にする部分に分けています。したがって0.1秒を５回数えるのを１０回にすれば１秒点灯、１秒消灯というように変更できます。

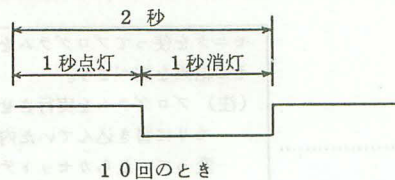
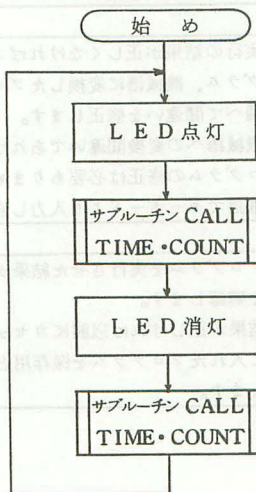
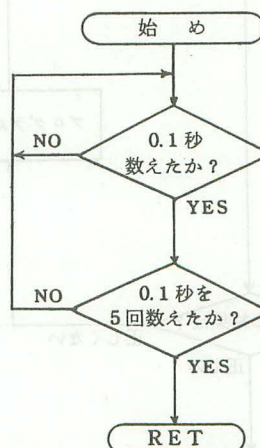



図3.4 フローチャート

(2) ソースプログラムを機械語に変換したリストを示します。

プログラムの開始アドレスは0000番地、終了アドレスは0018番地です。

```
            Z-80  ASSEMBLER  V1.1  PAGE   1

      アドレス   機械語                              アセンブリ言語
   1                                  ORG        0
   2  0000   D3DC         LOOP:     OUT        (0DCH),A
   3  0002   CD0C00                 CALL       TCOUNT
   4  0005   DBDC                   IN         A,(0DCH)
   5  0007   CD0C00                 CALL       TCOUNT
   6  000A   18F4                   JR         $-10
   7  000C   0605         TCOUNT:   LD         B,5
   8  000E   21E624       LOOP1:    LD         HL,24E6H
   9  0011   2B           LOOP2:    DEC        HL
  10  0012   7C                     LD         A,H
  11  0013   B5                     OR         L
  12  0014   20FB                   JR         NZ,$-3
  13  0016   10F6                   DJNZ       $-8
  14  0018   C9                     RET
```

ソースプログラムを機械語に変換するときは次の点に注意してください。

(a) 16ビットのデータをレジスタなどに設定するときの上位8ビットのデータと下位8ビットのデータの並びは次の例に示すようにします。

メモリには、21,00,24の順序で書き込みます。

(b) リラティブジャンプ("JR")などを使用するときの例を示します。

「JR　NZ,$-3」の$-3はアセンブラの命令で、「JR　NZ,$-3」のアドレス0023番地から-3番地(0020番地)へ戻ることを意味します。

ただし、機械語に変換するときは「JR　NZ，$－3」の次のアドレス0025番地を00とし、戻りたいアドレスまで－1を行います。

ディスプレイスメントがプラスのときは、次のアドレスを00とし、＋1を行います。

（例）

アドレス	アセンブリ言語	機械語
0020	JR　NZ，　$＋4	2003
0022	LD　A，　H	00…………7C
0023	LD　B，　01H	0102………0601
0024	LD　C，　05	03…………0E05

(3)　機械語に変換したプログラムをキーボードより入力します。

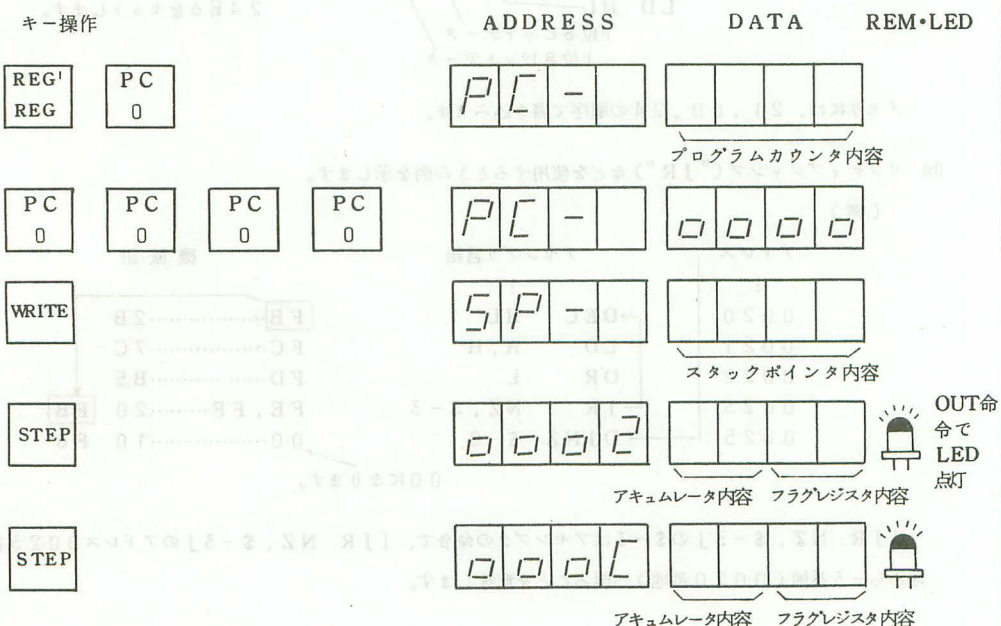
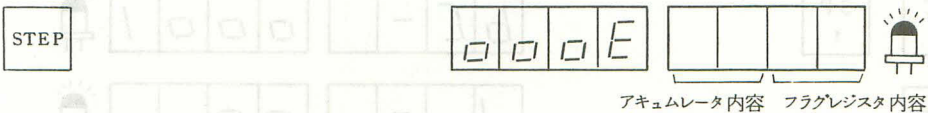
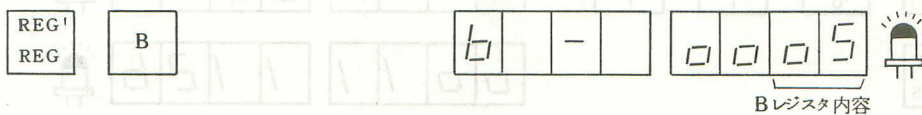
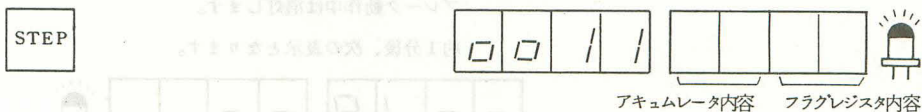
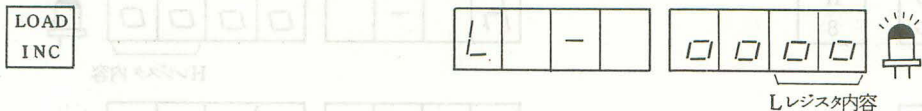
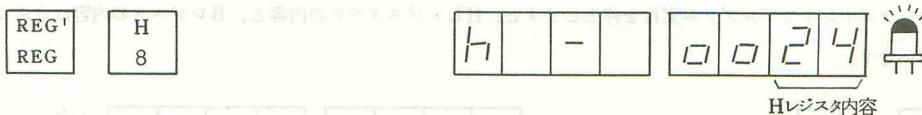
機械語に変換したプログラムをアドレス0000番地よりメモリに書き込んでください。

(4)　カセットテープにストアします。

(3)でメモリに書き込んだプログラムをカセットテープにストアしてください。プログラムはアドレス0000～0018番地のメモリに書き込まれています。

3.4.2　プログラムのデバックの仕方

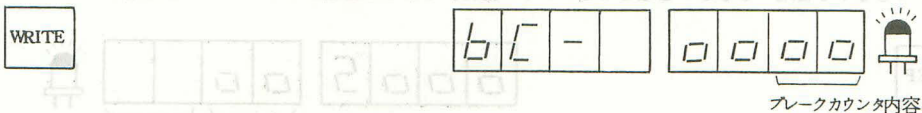
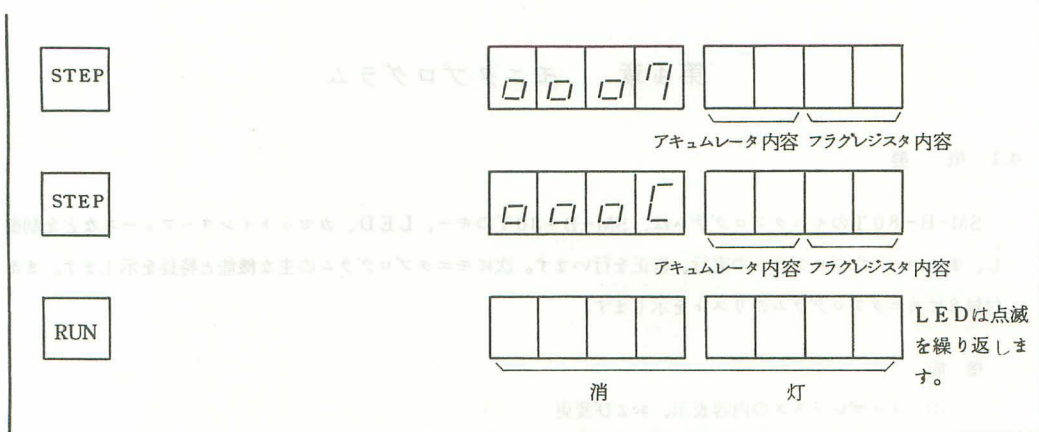
3.2.7項のステップ動作、3.2.8項のブレーク動作を使ってプログラムに誤りがないかを調べます。

（操作例）

ここで、アドレス000C番地の命令「LD　B，5」でBレジスタに16進数の05が書き込まれたかを確認します。

続いてステップ動作させます。

ここで、アドレス000E番地の命令「LD　HL，2400H」でHLレジスタペアに16進数の2400が書き込まれたかを確認します。

(2)　ブレーク動作

ブレークポイントを0018番地、繰り返し回数を1回としてアドレス0011番地よりブレーク動作させます。

ブレークポイントでプログラム実行を停止しますと、HLレジスタペアの内容と、Bレジスタの内容は0になっています。

続いて0018番地～0007番地までをステップ動作、000C番地からプログラムを実行させます。

STEP	0007	アキュムレータ内容　フラグレジスタ内容
STEP	000C	アキュムレータ内容　フラグレジスタ内容
RUN	消　灯	LEDは点滅を繰り返します。

SM-B-80Tで使用するモニタプログラムについては第4章で詳しく説明します。

第4章　モニタプログラム

4.1 概　要

SM-B-80Tのモニタプログラムは、SM-B-80Tのキー、LED、カセットインターフェースなどを制御し、またユーザ プログラムの実行、修正を行います。次にモニタプログラムの主な機能と特長を示します。また付録8にモニタプログラムのリストを示します。

機　能

① ユーザレジスタの内容表示、および変更

② メモリの内容表示、および変更

③ ユーザプログラムの実行

④ ユーザプログラムのステップ実行

⑤ カセットテープよりのロード、およびストア

⑥ ブレークポイント、ブレークカウンタ機能

特　長

① レジスタのシンボル表示

② 16ビットレジスタの変更が容易に行える

③ オーディオカセットのリモート操作

④ カンサスシティ規格採用

4.2 構　成

モニタプログラムは、次の三つの部分より構成されています。

① メインルーチン

② コマンド処理ルーチン

③ 割り込み処理ルーチン

メインルーチンは、キー入力を待ち、入力されたキーに従って処理を進めます。データキーが入力されると、そのデータをLEDに表示し、またコマンドキーが入力されると、そのコマンドの処理ルーチンへ制御を移します。

コマンド処理ルーチンは、各コマンドキーに対応した処理を行います。

割り込み処理ルーチンは、SM-B-80Tで使用しているノンマスカブルインターラプトの処理ルーチンで、レジスタの退避、ブレークカウンタの更新などを行います。

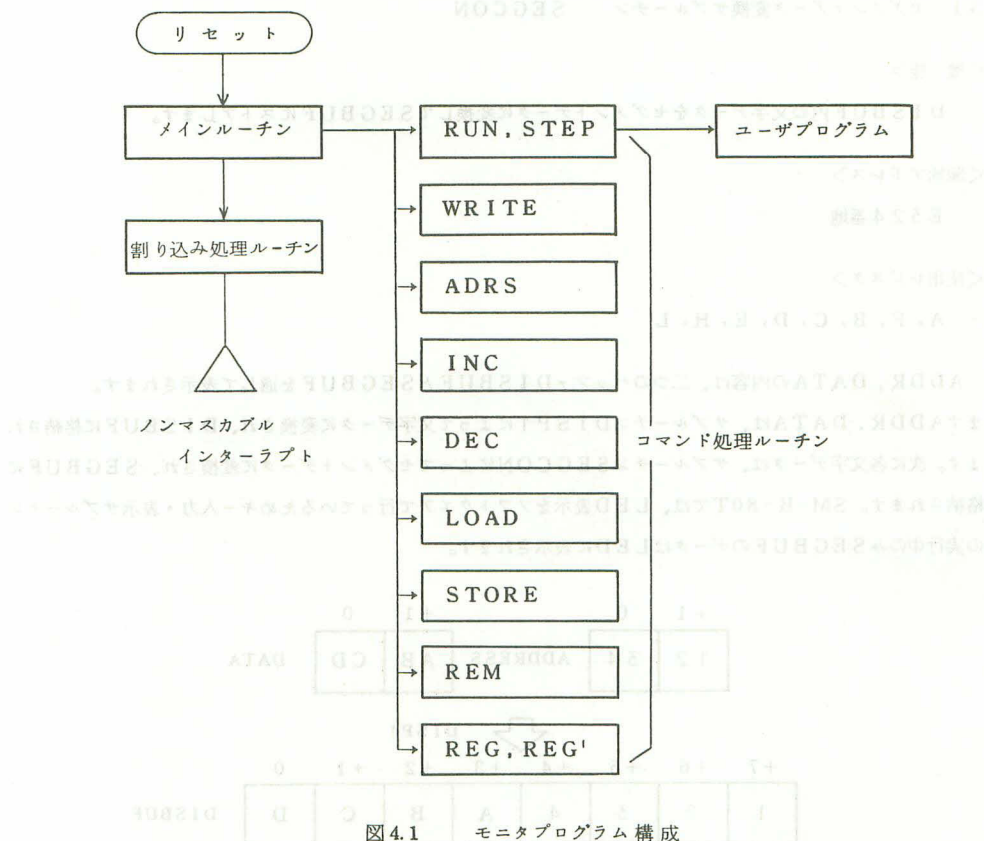



図4.1　モニタプログラム構成

4.3　モニタサブルーチン

モニタで使用しているいくつかのサブルーチンを、ユーザプログラムで使用することができます。モニタサブルーチンの名前と開始アドレスを次に示します。

名　前		アドレス
① セグメントデータ変換サブルーチン	（SEGCON）	E324
② 文字データ変換サブルーチン	（DISP）	E2F7
③ キー入力・LED表示サブルーチン	（KEYIN）	E33F
④ タイマーサブルーチン	（WAIT）	E2B7
⑤ カセットロードサブルーチン	（LOAD）	E1C6
⑥ カセットストアサブルーチン	（STORE）	E236
⑦ LED表示サブルーチン	（SCAN）	E370

以降の説明では、アドレスの代りに4.4項のメモリマップの名前を使用します。

4.3.1　セグメントデータ変換サブルーチン　　SEGCON

＜機　能＞

DISBUF内の文字データをセグメントデータに変換してSEGBUFにストアします。

＜開始アドレス＞

E324番地

＜使用レジスタ＞

A，F，B，C，D，E，H，L

ADDR，DATAの内容は、二つのバッファDISBUFとSEGBUFを通して表示されます。まずADDR，DATAは、サブルーチンDISP1によって文字データに変換され、DISBUFに格納されます。次に各文字データは、サブルーチンSEGCONによってセグメントデータに変換され、SEGBUFに格納されます。SM-B-80Tでは、LED表示をソフトウエアで行っているためキー入力・表示サブルーチンの実行中のみSEGBUFのデータはLEDに表示されます。

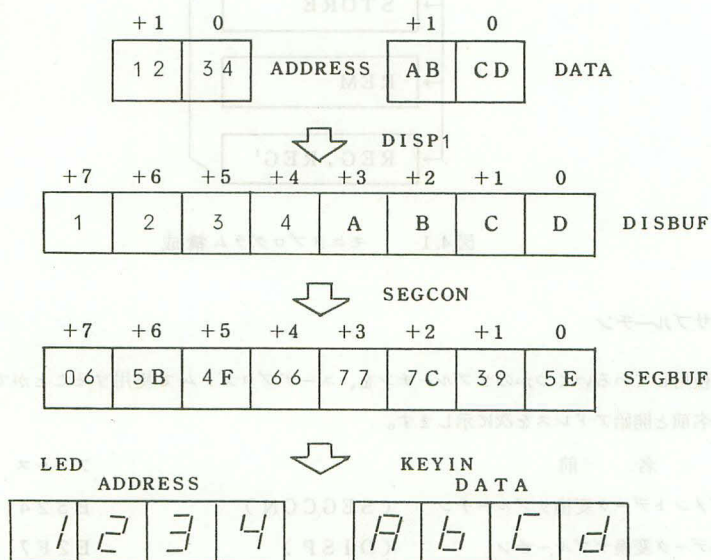



図4.2　表示データの変換

SM-B-80Tは、表4.1に示す26種の文字を表示することができます。各文字には順に番号が割り当ててあってこの番号をこの文字の文字データと呼びます。またLEDのどのセグメントを点灯させるかに対応したデータをこの文字のセグメントデータと呼びます。

LED

7	6	5	4	3	2	1	0
ビット7	ビット6	ビット5	ビット4	ビット3	ビット2	ビット1	ビット0

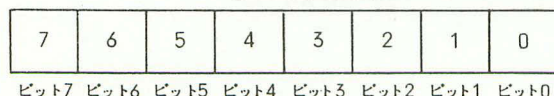
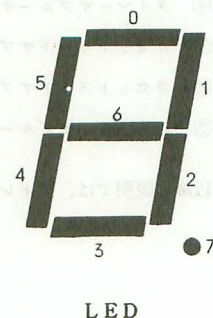



図4.3　セグメントデータ

文字	文字データ	セグメントデータ	LED
0	00	5C	
1	01	06	
2	02	5B	
3	03	4F	
4	04	66	
5	05	6D	
6	06	7D	
7	07	27	
8	08	7F	
9	09	6F	
A	0A	77	
B	0B	7C	
C	0C	39	
D	0D	5E	
E	0E	79	
F	0F	71	
H	10	74	
L	11	38	
P	12	73	
X	13	76	
Y	14	6E	
—	15	40	
'	16	20	
空白	17	00	
I	18	06	
S	19	6D	

表 4.1　文字表示変換表　（16進数）

<フローチャート>

図4.4　セグメント変換サブルーチン

<注　意>

SEGCONは、デシマルポイントをサポートしていません。

デシマルポイントを点灯するには、SEGBUFの対応するバイトのビット7を1にしなければなりません。

<例>

LEDに1～8の数を表示するルーチンです。

```
   アドレス  機械語                  アセンブリ言語
1                                   ORG    0
2  0000    21D5FF                   LD     HL,DISBUF
3  0003    0608                     LD     B,8
4  0005    70              LOOP:    LD     (HL),B
5  0006    23                       INC    HL
6  0007    10FC                     DJNZ   LOOP
7  0009    CD24E3                   CALL   SEGCON
8  000C    CD3FE3                   CALL   KEYIN
```

4.3.2　文字データ変換サブルーチン　　DISP

＜機　能＞

ADDR，DATAの内容を文字データに変換してDISBUFにストアし、SEGCONをコールしてセグメントデータをSEGBUFにストアします。

＜開始アドレス＞

E2F7番地

＜使用レジスタ＞

A，F，B，C，D，E，H，L，B'，D'，E'，H'，L'

＜フローチャート＞

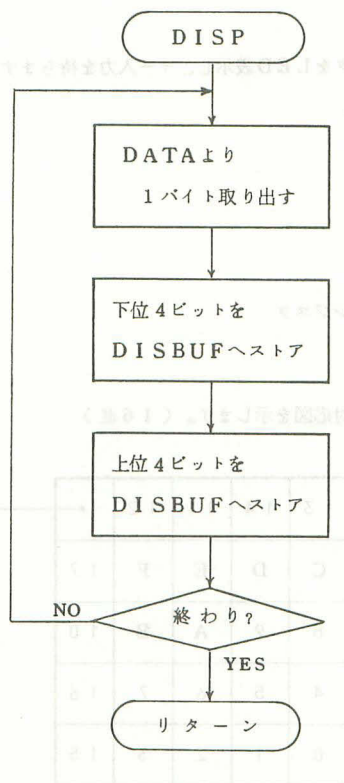



図4.5　文字データ変換サブルーチン

<例>

ＬＥＤに１～８の数を表示するルーチンです。

```
          アドレス   機械語              アセンブリ言語
  9                                ORG     0
 10       0000     213412          LD      HL,1234H
 11       0003     22E1FF          LD      (ADDR),HL
 12       0006     217856          LD      HL,5678H
 13       0009     22DFFF          LD      (DATA),HL
 14       000C     CDF7E2          CALL    DISP
 15       000F     CD3FE3          CALL    KEYIN
```

4.3.3　キー入力・ＬＥＤ表示サブルーチン　　ＫＥＹＩＮ

<機　能>

ＳＥＧＢＵＦ内のセグメントデータをＬＥＤ表示し、キー入力を待ちます。キー入力があればそのキー番号をＡレジスタに入れてリターンします。

<開始アドレス>

Ｅ３３Ｆ番地

<使用レジスタ>

Ａ'，Ｆ'，Ｃ'，ＩＸ，ＩＹを除く全レジスタ

<出　力>

図4.6にキーの位置とキー番号の対応図を示します。（１６進）

13	14	11	12	● (RESET キー)
C	D	E	F	17
8	9	A	B	10
4	5	6	7	16
0	1	2	3	15

図4.6　キー番号

<例>

入力したキーのキー番号をLEDに表示します。

	アドレス	機械語		アセンブリ言語	
16				ORG	0
17	0000	CD3FE3	LOOP :	CALL	KEYIN
18	0003	6F		LD	L, A
19	0004	2600		LD	H, 0
20	0006	22DFFF		LD	(DATA), HL
21	0009	CDF7E2		CALL	DISP
22	000C	18F2		JR	LOOP

4.3.4 タイマーサブルーチン　WAIT

<機　能>

Bレジスタの内容をnとすると、n×0.5秒後にリターンします。

<開始アドレス>

E2B7番地

<使用レジスタ>

A, F, B, H, L

<フローチャート>

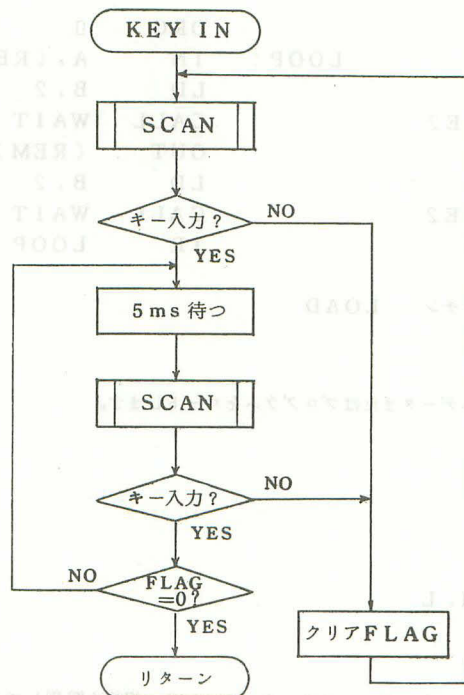



図4.7　キー入力・LED表示サブルーチン

<フローチャート>

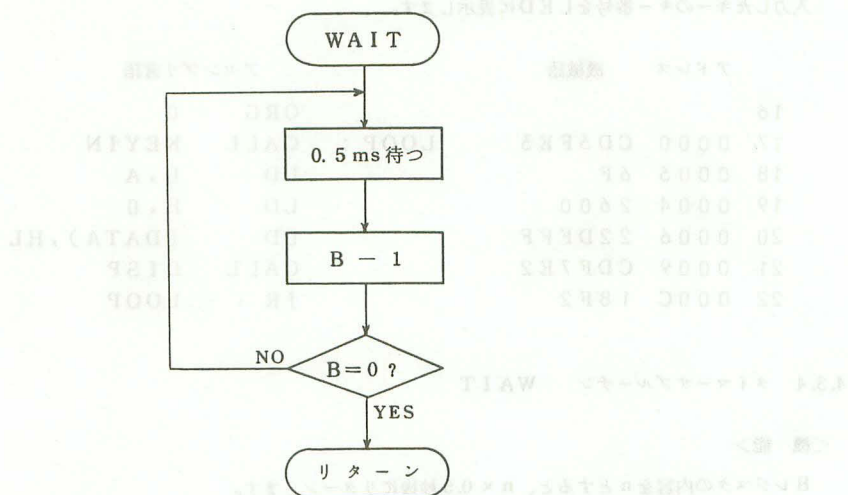



図 4.8　タイマーサブルーチン

<例>

1秒ごとにREM用LEDが点滅します。

	アドレス	機械語		アセンブリ言語	
23				ORG	0
24	0000	DBDC	LOOP :	IN	A, (REM)
25	0002	0602		LD	B, 2
26	0004	CDB7E2		CALL	WAIT
27	0007	D3DC		OUT	(REM), A
28	0009	0602		LD	B, 2
29	000B	CDB7E2		CALL	WAIT
30	000E	18F0		JR	LOOP

4.3.5　カセットロードサブルーチン　　LOAD

<機　能>

カセットテープよりメモリへデータまたはプログラムをロードします。

<開始アドレス>

E1C6番地

<使用レジスタ>

A, F, B, C, D, E, H, L

<データフォーマット>

SM-B-80 Tのカセットインターフェースは、カンサスシティ規格を採用しており、次に示す仕様となっています。

モード	直列非同期
伝送速度	３００ボー
マーク（論理“１”）	２４００Hz を８サイクル
スペース（論理“０”）	１２００Hz を４サイクル
テープヘッダ	テープの開始よりマーク周波数を３０秒間記録
ブロックヘッダ	マーク周波数を５秒間記録
スタートビット	１ビット（論理“０”）
ストップビット	２ビット（論理“１”）
データビット	８ビット

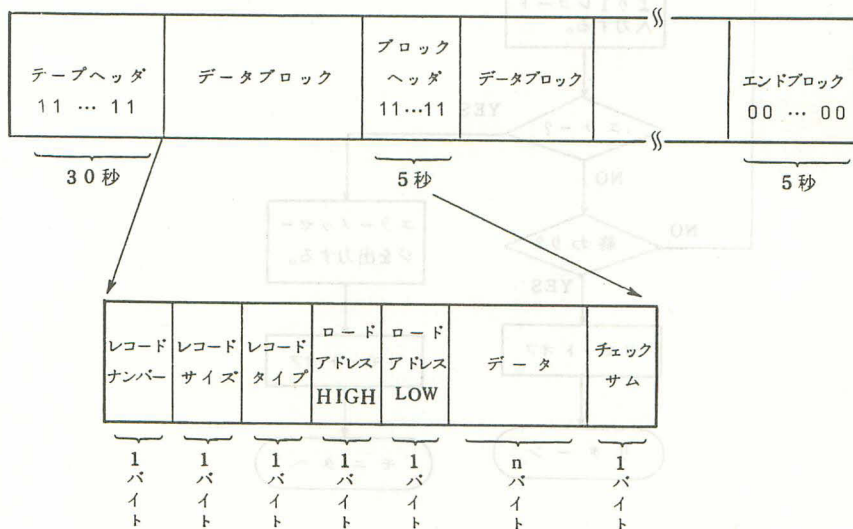



図4.9　カセットテープデータフォーマット

レコードナンバーは、レコードのシークェンスナンバを示す。１～最大２５５

レコードサイズは、　このレコードのデータ長を示す。　　最大２５５バイト

レコードタイプは、　このレコードの種類を示す。“０”の場合、エンドレコード

“１”の場合、データレコードです。

<注　意>

もしロード中にエラーを生じると、ＬＥＤへ“－－－－－－－－－ ”を表示してモニタへ戻ります。

<フローチャート>

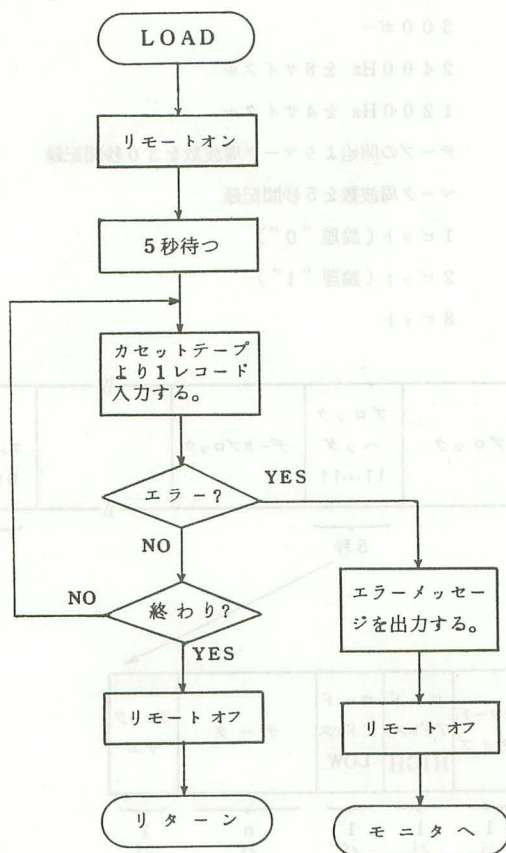



図4.10　カセットロードサブルーチン

4.3.6　カセットストアサブルーチン　　STORE

<機　能>

カセットテープへデータ、または、プログラムをストアします。

このルーチンをコールする前にストアするメモリブロックの開始アドレスをADDRへ、最終アドレスをDATAにセットしていなければなりません。

<開始アドレス>

E236番地

<使用レジスタ>

A，F，B，C，D，E，H，L

<データフォーマット>

4.3.5　参照

<例>

0番地より100番地の内容をカセットへストアします。

	アドレス	機械語	アセンブリ言語	
31			ORG	0
32	0000	210000	LD	HL，0
33	0003	22E1FF	LD	(ADDR)，HL
34	0006	210001	LD	HL，100H
35	0009	22DFFF	LD	(DATA)，HL
36	000C	CD36E2	CALL	STORE

<フローチャート>

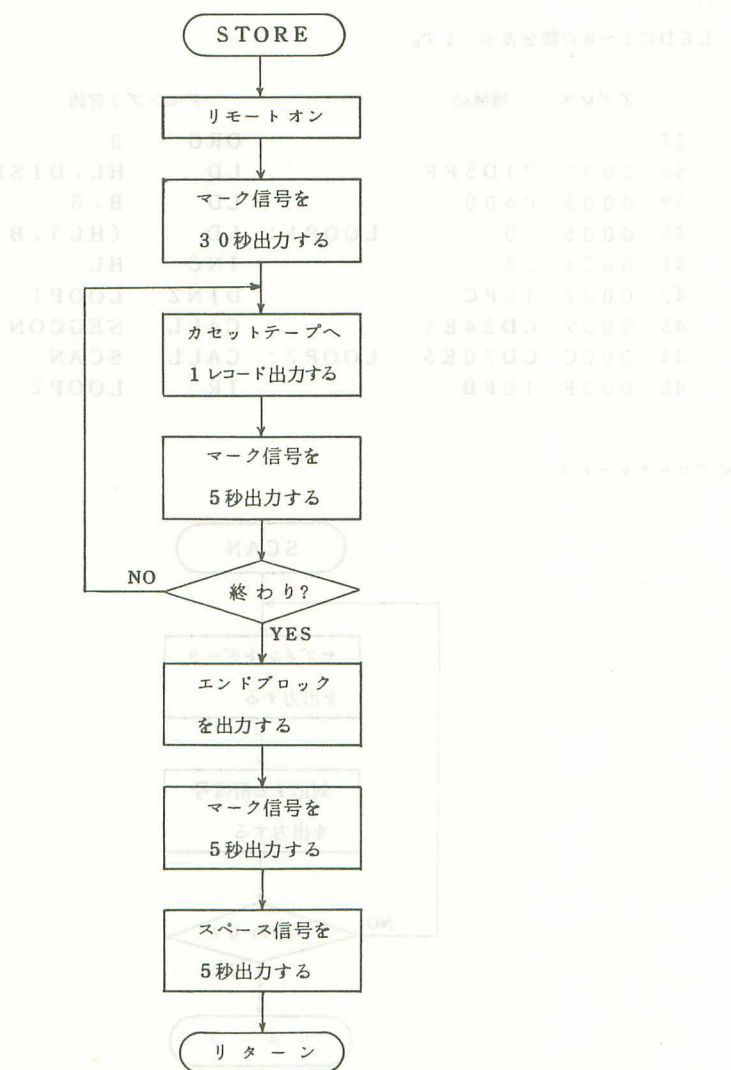



図4.11　カセットストアサブルーチン

4.3.7 LED表示サブルーチン　SCAN

<機　能>

SEGBUF内のセグメントデータをLEDに表示します。

SM-B-80TではLEDの表示をダイナミックスキャンで行っているため続けてコールしてください。

<開始アドレス>

E370番地

<使用レジスタ>

A，F，C，D，H，L

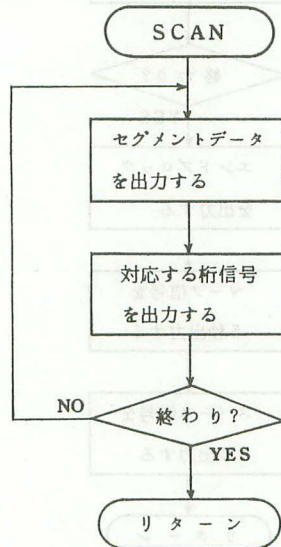
<例>

LEDに1～8の数を表示 ます。

```
      アドレス  機械語              アセンブリ言語
37                              ORG    0
38    0000     21D5FF           LD     HL,DISBUF
39    0003     0608             LD     B,8
40    0005     70      LOOP1:   LD     (HL),B
41    0006     23               INC    HL
42    0007     10FC             DJNZ   LOOP1
43    0009     CD24E3           CALL   SEGCON
44    000C     CD70E3  LOOP2:   CALL   SCAN
45    000F     18FB             JR     LOOP2
```

<フローチャート>

図4.12　LED表示サブルーチン

4.4 モニタワーキングエリアのメモリマップ

SM-B-80Tは、FF00番地よりFFFF番地までの256バイトのモニタ専用RAMを持っています。このRAMはモニタのワーキングエリアとして用いられます。

アドレス			
FFFF	BADDR	H	ブレークアドレスレジスタ H
FFFE		L	L
FFFD	BCOUNT		ブレークカウンタ
FFFC	PC	H	
FB		L	
FA	A		
F9	F		
F8	I		
F7	IF		
F6	B		
F5	C		
F4	D		
F3	E		
F2	H		
F1	L		
F0	A'		ユーザレジスタ
EF	F'		セーブエリア
EE	B'		
ED	C'		
EC	D'		
EB	E'		
EA	H'		
E9	L'		
E8	IX	H	
E7		L	
E6	IY	H	
E5		L	
E4	SP	H	
FFE3		L	
FFE2	ADDR	H	アドレスレジスタ
E1		L	
FFE0	DATA	H	データレジスタ
DF		L	
FFDE	REMSW		リモートスイッチフラグ
FFDD	FLAG		キーインプットフラグ
FFDC	DISBUF	8	ディスプレイバッファ
DB		7	
DA		6	
D9		5	
D8		4	
D7		3	
D6		2	
FFD5		1	

アドレス	名前		内容
FFD4	SEGBUF	8	セグメントバッファ
D3		7	
D2		6	
D1		5	
D0		4	
CF		3	
CE		2	
FFCD		1	
FFCC	MODE		モードフラグ　STEP／RUN
FFCB ≈ FF9A	STACK		モニタスタックエリア
FF99 ≈ FF00	USER		ユーザースタックエリア

表 4.2　モニタワーキングエリア

ユーザスタックとして１５４バイトのエリアが確保されています。スタックポインタはリセットするたびにこのエリアを指すよう設定されますので、サブルーチンネスティングは８７まで可能です。

第5章　SM-B-80Tハードウェア

この章では、SM-B-80Tの構成、ハードウエアについて説明します。

なお、LSIについては付属のLSI資料を参照してください。

5.1　マイクロコンピュータの基本的な構成

マイクロコンピュータは、基本的にはCPU、メモリ、I／Oポートで最小のシステムが構成できます。

図5.1にZ－80　CPUを使用するマイクロコンピュータの基本的な構成を示します。

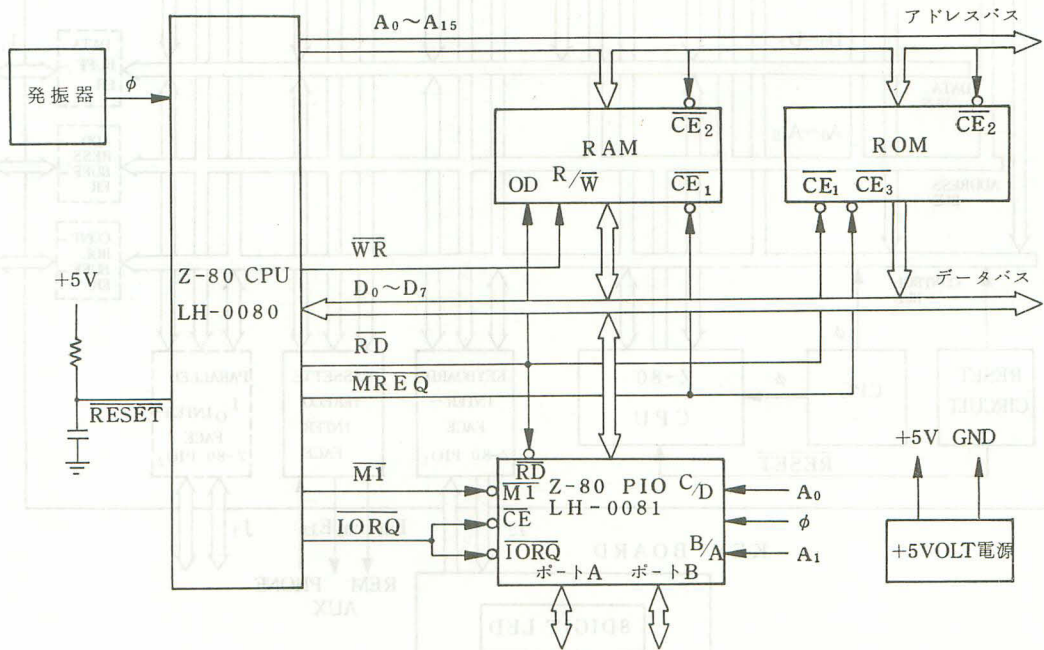



図5.1　　基　本　的　な　構　成

マイクロコンピュータのシステムとしては一般的に経済性、信頼性から制御プログラム（モニタプログラム）をROMに書いておき、RAMは、データの保存場所として、また、スタックとして使用しています。

Z-80　CPUは5V単一電源で動きますので、ROM、RAMにも5V単一電源のものを使えば電源の種類が少なくてすみ、電源構成が簡単になります。

コンピュータシステムには、コンピュータ内部と外部世界とを結ぶ（データ交信）I／O回路が必要です。

図5.1の構成では、I／Oポートとして Z－80　PIOを使用しています。

Z－80　PIOは、CPUから送られてきたデータを外部に出力したり、また、外部から入力されたデータをコンピュータ内部のデータバスに取り込んだりします。

図5.1はマイクロコンピュータの基本的な構成ですので、メモリなどのアドレス配分はアドレス信号を直接使用していますが、将来メモリなどの拡張を考えた場合、アドレスデコードしてアドレス配分をした方が非常に有利となります。

5.2　SM－B－80Tのシステム構成

図5.2にSM－B－80Tのシステム構成を示します。

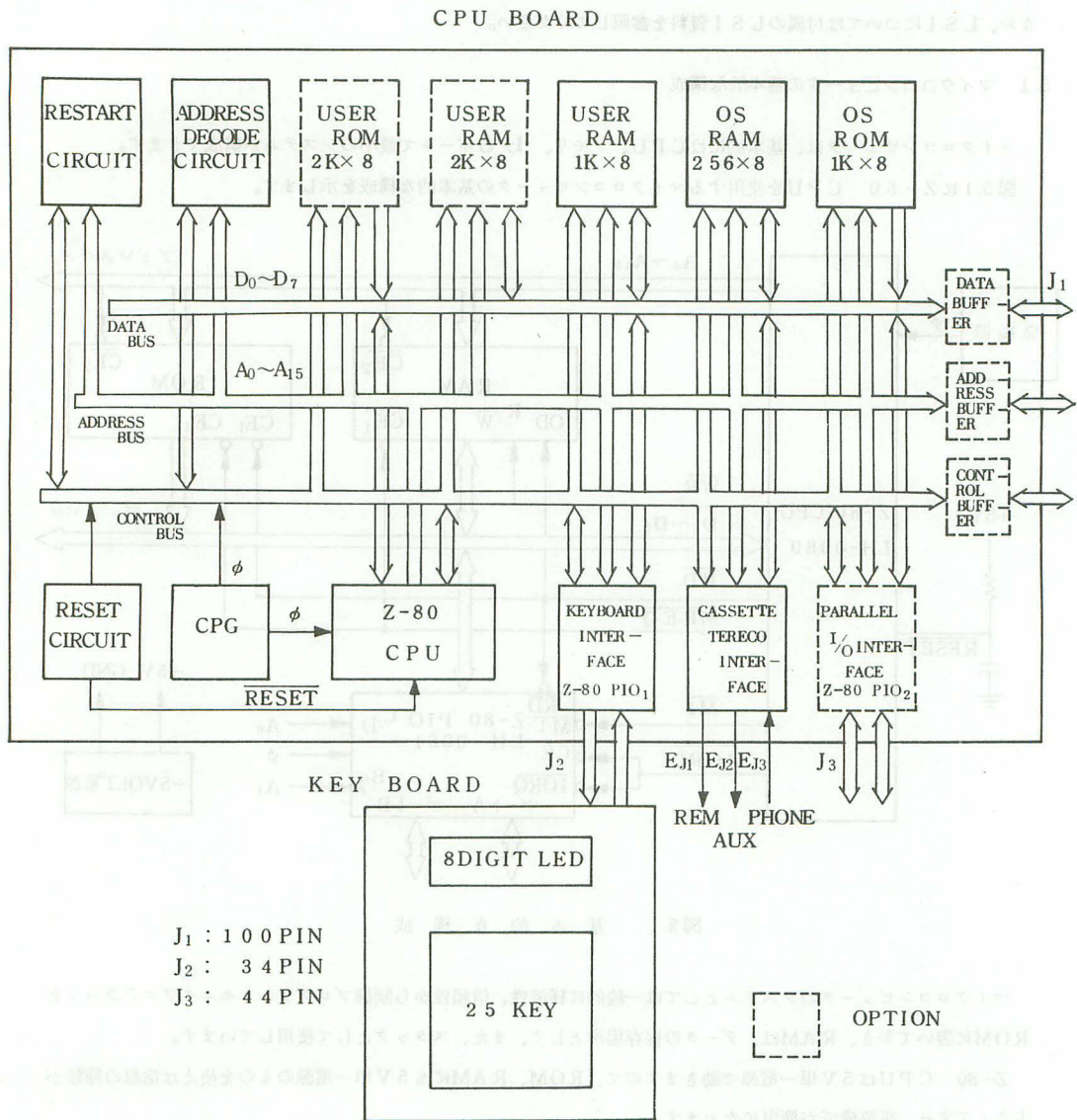



図5.2　SM－B－80Tのシステム構成

SM－B－80TはCPUにLH－0080（Z－80　CPU）を使用しています。

ROMにはバイポーラのLH－7055（1Kバイト）を使用し、モニタプログラムを書き込んで1個実装しています（最大2個まで搭載できます）。

RAMにはモニタプログラム専用のLH－2111A4（256バイト）と、ユーザ用としてLH－2114－3（1Kバイト）を使用しています。また、LH－2114－3は最大3Kバイトまで拡張できます。さらに、メ

モリのチップ選択はアドレスデコード回路でアドレス信号A_8～A_{15}をデコードした信号を使用しています。

オーディオカセットインターフェースは、８２５１（ＵＡＲＴ）を使用してのデータの並列、直列変換回路と変調回路、復調回路、リモート回路、分周回路より構成されます。

パラレルのＩ／ＯポートはＬＨ－００８１（Ｚ－８０　ＰＩＯ）を使用しています（オプション）。この他に、キーボードとのインターフェースに使用しているＬＨ－００８１もキーボードを使用しなければパラレルＩ／Ｏポートとして使用できます。

キーボードは、２５キー、８桁７セグメントＬＥＤ、デコード回路、ＬＥＤ駆動回路より構成し、３４芯のフラットケーブル（J_2）でＣＰＵボードと接続しています。

アドレス、データ、コントロール信号はバッファを介して１００ピンコネクタ（J_1）へつながっていますので、バッファを取り付けることによりJ_1を用いてシステムを拡張できるようになっています。

SM-B-80Tはこの他に、リスタートアドレス変換回路、リセット回路、水晶発振回路などにより構成されています。

図5.3にSM-B-80Tのシステム構成の詳細を示します。

5.3　アドレス配置

SM-B-80Tのメモリアドレス、ポートアドレスについて説明します。

5.3.1　メモリアドレス

表5.1にSM-B-80Tのメモリアドレスを示します。

メモリのアドレス配置は、アドレス信号A_{15}～A_8をデコードして各メモリのチップ選択信号として使用しています。

ROMは、アドレスＥ０００番地～Ｅ７ＦＦ番地の２Ｋバイトに配置しています。

モニタプログラムが使用するＲＡＭは、アドレスＦＦ００番地～ＦＦＦＦ番地の２５６バイトに配置しています。

ユーザ用ＲＡＭは、アドレス００００番地～０ＢＦＦ番地の３Ｋバイトに配置しています。

5.3.2　ポートアドレス

表5.2にSM-B-80Tのポートアドレスを示します。

ポートアドレスの配置は、アドレス信号A_7～A_0をデコードして各Ｉ／Ｏポートの選択信号として使用しています。

ポートアドレスのＤＥ～ＦＦは将来のシステム拡張用としています。

4.9152MHz
CPG
STEP CIRCUIT
STEP
ADRS
NMI
LH−0080
Z−80 CPU
RESET
ADDRESS CONVERTER
ARES
E000
0000
FREQUENCY DIVIDER
RESET CIRCUIT
DATA BUS
CONTROL BUS
ADDRESS BUS
LH−7055 PROM 1KW×8
LH−2111A4 RAM 256 W×4 OD R/W
LH−2111 A4 RAM 256 W×4 OD R/W
LH−2114−3 RAM 1KW×4
LH−2114 RAM 1KW×4
OPTION
DATA BUFFER
CONTROL BUFFER
ADDRESS BUFFER
C/D 8251 UART
TXC
TXD RXD RXC
4.8kHz
1.2kHz
2.4kHz
614.4kHz
CASSETTE TERECO INTERFACE
REM AUX PHONE
C/DB/A CE LH−0081 Z−80PIO1
DECODER
LED
25 KEY
KEY BOARD
MEMORY ADDRESS DECODER
PORT ADDRESS DECODER
PORT A LH−0081 Z−80PIO2 PORT B
OPTIN
FF DE システム予備
DD U44 F/F
DC REM
DB DA 8251
D9 STEP
D8 ARES
D7 D4 PIO2 OPTION
D3 D0 PIO1
CF 00
ポートアドレス
FFFF FF00 256B RAM
E7FF E400 1KB ROM OPTION
E3FF E000 1KB ROM
0BFF 0400 2KB RAM OPTION
03FF 0000 1KB RAM
メモリアドレス

図5.3 システム構成

アドレス	容量	メモリ	内容
FFFF FF00	256	RAM	LH−2111A4 （ワーキングエリア）
FEFF E800	5K + 768	—	ブランク
E7FF E400	1K	ROM	LH−7055 （オプション）
E3FF E000	1K	ROM	LH−7055 （モニタプログラム）
DFFF 0C00	53K	—	ブランク
0BFF 0400	2K	RAM	LH−2114 （オプション）
03FF 0000	1K	RAM	LH−2114−3 （ユーザエリア）

表5.1　メモリアドレス

アドレス	内容	
FF DE	システム予備	
DD	U_{44} F/F	セット　（OUT命令） リセット（IN命令）
DC	リモート端子	ON（OUT命令） OFF（IN命令）
DB DA	8251	コントロール データ
D9	$\overline{\text{STEP}}$	
D8	$\overline{\text{ARES}}$	
D7 D6 D5 D4	PIO2	ポートBコントロール ポートBデータ ポートAコントロール ポートAデータ
D3 D2 D1 D0	PIO1	ポートBコントロール ポートBデータ ポートAコントロール ポートAデータ
CF 00	ブランク	

表5.2　ポートアドレス

5.4　リスタート回路

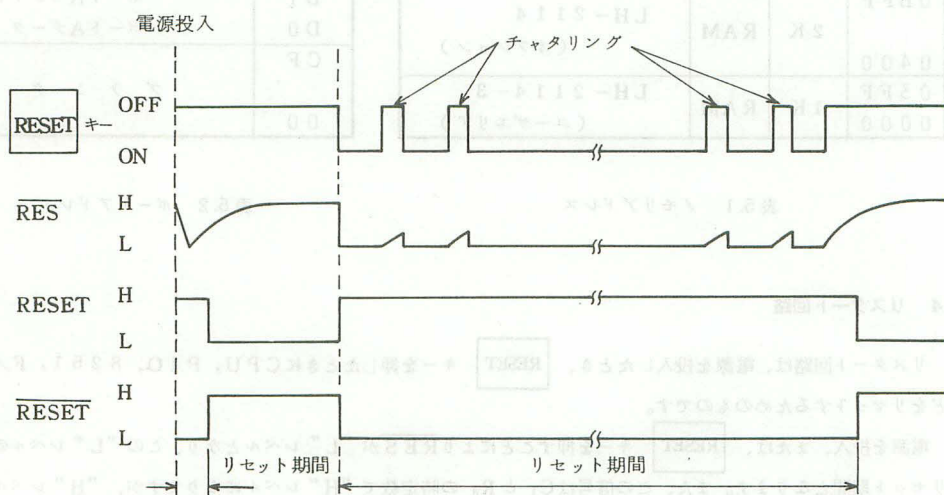
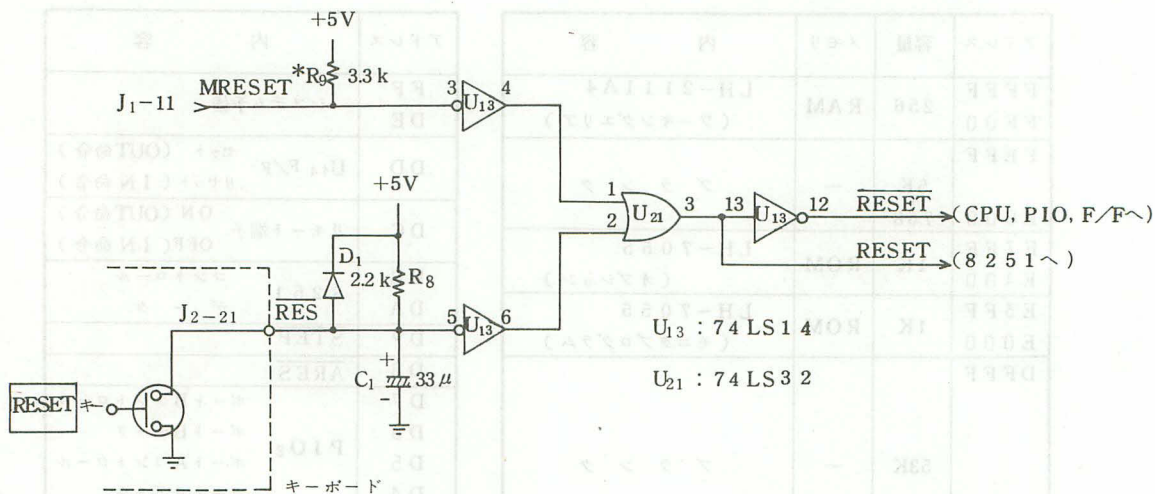
リスタート回路は、電源を投入したとき、RESET キーを押したときにCPU，PIO，8251，F/Fなどをリセットするためのものです。

電源を投入、または、RESET キーを押すことにより$\overline{\text{RES}}$が“L”レベルとなり、この“L”レベルの間がリセット期間となります。また、この信号はC_1とR_8の時定数で“H”レベルになりますが、“H”レベルになるとリスタートします。

CPU，PIO，F/Fに対しては$\overline{\text{RESET}}$が、8251に対してはRESETがリセット信号になります。また、$\text{MR\dot{E}SET}^{*}$は外部よりリセット信号を入力するラインです。

図5.4にリスタート回路を示します。また、図5.5にタイミングを示します。

図5.4　リスタート回路

図5.5　電源投入および手動リスタートのタイミング波形

RESET キーを押したときのチャタリングはC_1とR_8の時定数を充分に大きくして防いでいます。

5.5　アドレス変換回路

アドレス変換回路は、アドレス切り換えスイッチでCPUのリスタートアドレスをユーザプログラムの開始アドレス0000番地か、モニタプログラムの開始アドレスE000番地のどちらかに変換するためのものです。

図5.6にアドレス変換回路を示します。

Z－80　CPUのリスタートアドレスは００００番地ですが、アドレス切り換えスイッチをＥ０００側に設定しておきますと$\overline{\text{RESET}}$により７４７４（U_{47}）が“H”にセットされますので、CPUのアドレス信号A_{15}～A_{13}を“H”（1）にします。これによりアドレスがＥ０００番地に変換されます。

モニタプログラムをＥ０００番地より実行しますと次の命令が書き込まれていますのでCPUのプログラムカウンタ（PC）の書き換えと、７４７４（U_{47}）を“L”にリセットするためアドレス信号A_{15}～A_{13}を“H”（1）にするのを中止します。命令の詳細は付録８のモニタプログラムリストを参照してください。

図5.6　アドレス変換回路

5.6　ステップ回路

ステップ回路は、シングルステップ機能によりプログラムの中の１命令を実行したときに$\overline{\text{NMI}}$（ノンマスカブルインターラプト）を発生させるためのものです。

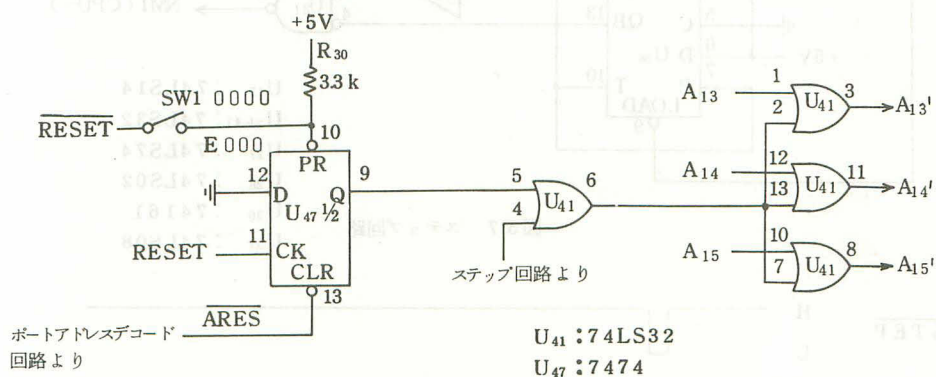



図5.7にステップ回路を、図5.8にステップ回路のタイミング波形を示します。

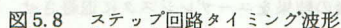
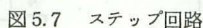
[STEP]　キーを押しますとCPUがSTEP（ポートアドレスＤ９）を“L”レベルにします。この$\overline{\text{STEP}}$で７４ＬＳ７４（U_{27}－½）を“H”にセットしてカウンタ７４１６１（U_{36}）をカウント可能な状態にします。また、同時に７４１６１（U_{36}）にＤ，Ｃ，Ｂ，Ａの入力状態（１０１１）をセットします。

プログラムの１命令を実行してカウンタが１０１１になると$\overline{\text{NMI}}$を発生してCPUに送り、CPUはこれに対してアドレス信号にノンマスカブルインターラプトのアドレス００６６番地を出力します。これをステップ動作の処理ルーチンＥ０６６番地に変換しています。

処理ルーチンでは、$\overline{\text{ARES}}$（ポートアドレスＤ８）を“L”レベルにして７４ＬＳ７４（U_{27}－½）と74161（U_{36}）をリセットし、１命令実行後の結果をユーザレジスタに退避させ、次の命令が入っているアドレスをアド

レス表示部に、アキュムレータとフラグレジスタの内容をデータ表示部に表示する処理を行っています。

図 5.7　ステップ回路

図 5.8　ステップ回路タイミング波形

5.7　アドレスデコード回路

アドレスのデコード回路には、メモリのチップ選択用信号を発生させるメモリアドレスデコード回路と、ＰＩＯ，８２５１のチップ選択信号や、ステップ回路などに使用する制御信号を発生させるポートアドレスデコード回路があります。

5.7.1　メモリアドレスデコード回路

図5.9にメモリアドレスデコード回路を示します。

ユーザＲＡＭ領域のデコード回路では、アドレス００００番地〜１ＦＦＦ番地を1Kバイト単位でデコードしていますが、０Ｃ００番地〜１ＦＦＦ番地の５Ｋバイトは使用していません。

ＲＯＭ領域は、アドレスＥ０００番地〜ＦＦＦＦ番地を1Kバイト単位でデコードしていますが、Ｅ８００番地〜ＦＦＦＦ番地の５Ｋバイトは使用していません。ただし、ＦＦ００番地〜ＦＦＦＦ番地の２５６バイトはモニタプログラム専用のＲＡＭ領域（ＬＨ－２１１１Ａ４）に使用しています。

メモリのアドレスについては表5.1を参照してください。

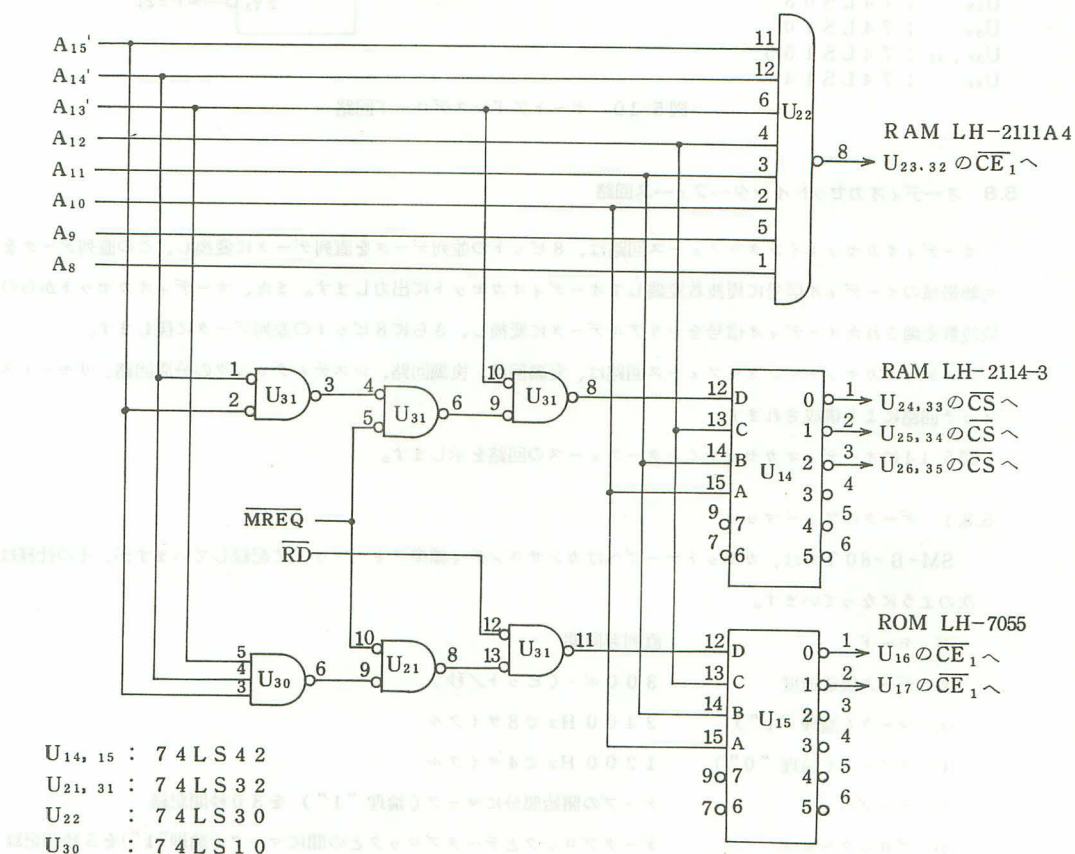



図5.9　メモリアドレスデコード回路

5.7.2　ポートアドレスデコード回路

図5. 10にポートアドレスデコード回路を示します。

ポートアドレスはD0～DFまでデコードしていますが、このうちモニタプログラムがD0～DDまでの14ポートを使用しています。

アドレスの詳細については表5.2を参照してください。

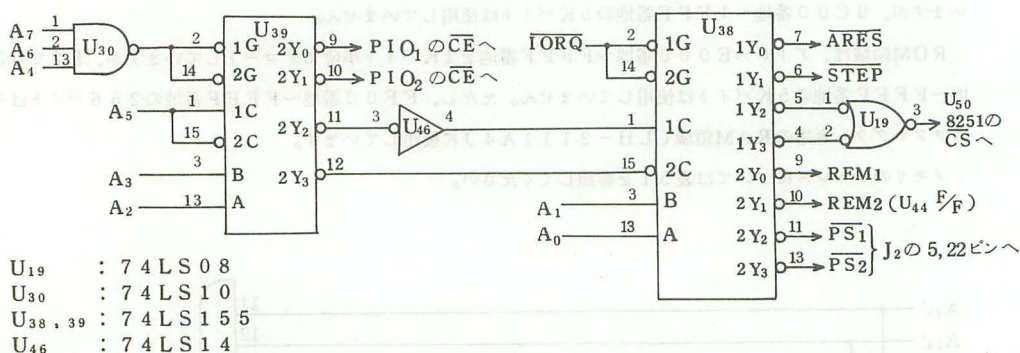



図5. 10　ポートアドレスデコード回路

5.8　オーディオカセットインターフェース回路

オーディオカセットインターフェース回路は、8ビットの並列データを直列データに変換し、この直列データを可聴帯域のオーディオ信号に周波数変調してオーディオカセットに出力します。また、オーディオカセットからの周波数変調されたオーディオ信号をシリアルデータに変換し、さらに8ビットの並列データに戻します。

オーディオカセットインターフェース回路は、変調回路、復調回路、システムクロックの分周回路、リモートスイッチ回路により構成されます。

図5. 14にオーディオカセットインターフェースの回路を示します。

5.8.1　データのフォーマット

SM-B-80Tでは、カセットテープへはカンサスシティ標準フォーマットで記録していますが、その仕様は次のようになっています。

(1) モード	直列非同期
(2) データ伝送速度	300ボー（ビット／秒）
(3) マーク（論理“1”）	2400Hzで8サイクル
(4) スペース（論理“0”）	1200Hzで4サイクル
(5) テープヘッダ	テープの開始部分にマーク（論理“1”）を30秒間記録
(6) ブロックヘッダ	データブロックとデータブロックとの間にマーク（論理“1”）を5秒間記録
(7) データブロック長	255バイト
(8) スタートビット	1ビット（論理“0”）

(9) ストップビット　　　　2ビット(論理"1")

(10) データビット　　　　8ビット

(11) エンドブロック　　　　終わりの部分にスペース(論理"0")を5秒間記録

CPUからの8ビット並列データは8251(UART)で直列データに変換し、自動的にスタートビットを1ビット、ストップビットを2ビット付加します。

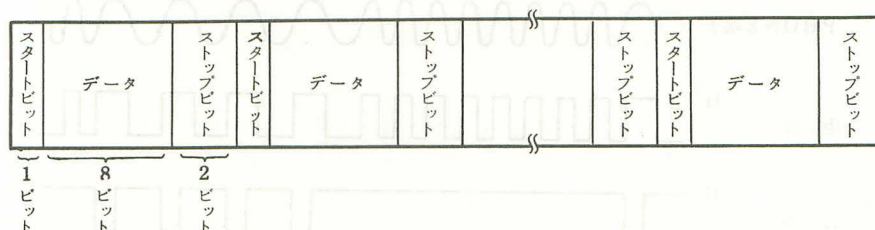



図5.11　データフォーマット

データフォーマットの詳細は4.3.5項のカセットロードサブルーチンを参照してください。

5.8.2　変調回路

変調回路では、8251(UART)からのシリアルデータを論理"0"のとき1200Hz、論理"1"のとき2400Hzに周波数変調します。

さらに、この変調したデータをフィルタと増幅器を通してAUX端子へ出力します。

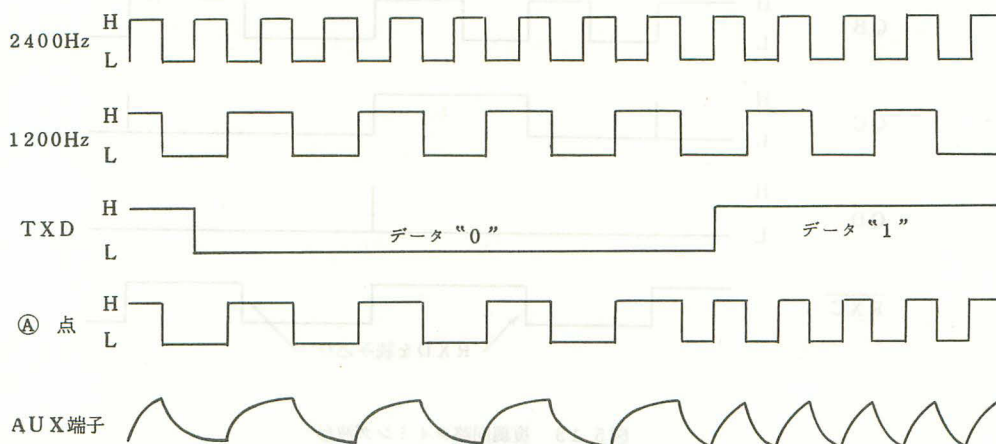



図5.12　変調回路タイミング波形

5.8.3　復調回路

復調回路では、カセットテープに記録したデータをカセットのイヤホン端子からCPUボードのPHONE端子に入力し、1200Hzをデータ"0"、2400Hzをデータ"1"に変換します。

PHONE端子に入力したデータは増幅され74LS14(U_{46})でパルスに波形整形されます。この整形された信号とクロック(614.4kHz)でカウント回路を働かせます。

カウント回路では、Ⓑ点の入力信号（データ）が１２００Hz のときはパルスになり、２４００Hz のときは“1” になるように設定していますのでⒸ点のような波形になります。

このⒸ点の信号から８２５１（U_{50}）のＲＸＤ（レシーブデータ）と$\overline{RXC}$（レシーブクロック）を作っています。また、８２５１（U_{50}）は$\overline{RXC}$の立ち上がりでＲＸＤを読み込みます。

図5. 13　復調回路タイミング波形

$U_{10,19}$	：74LS08	$U_{20,44,45}$	：74LS74	U_{46}	：74LS14
U_{11}	：74LS32	U_{28}	：74LS02	U_{48}	：17902
U_{12}	：74LS293	$U_{29,40}$	：4040	U_{50}	：8251
U_{18}	：7438	U_{30}	：74LS10		

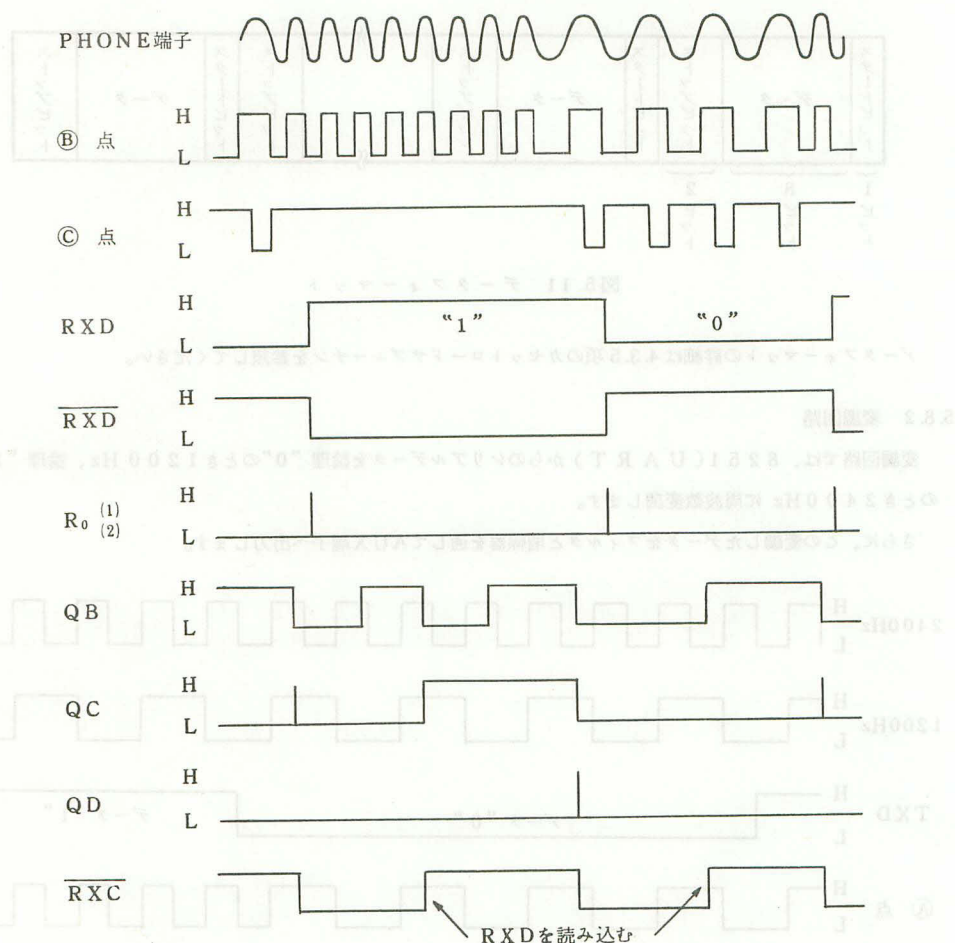
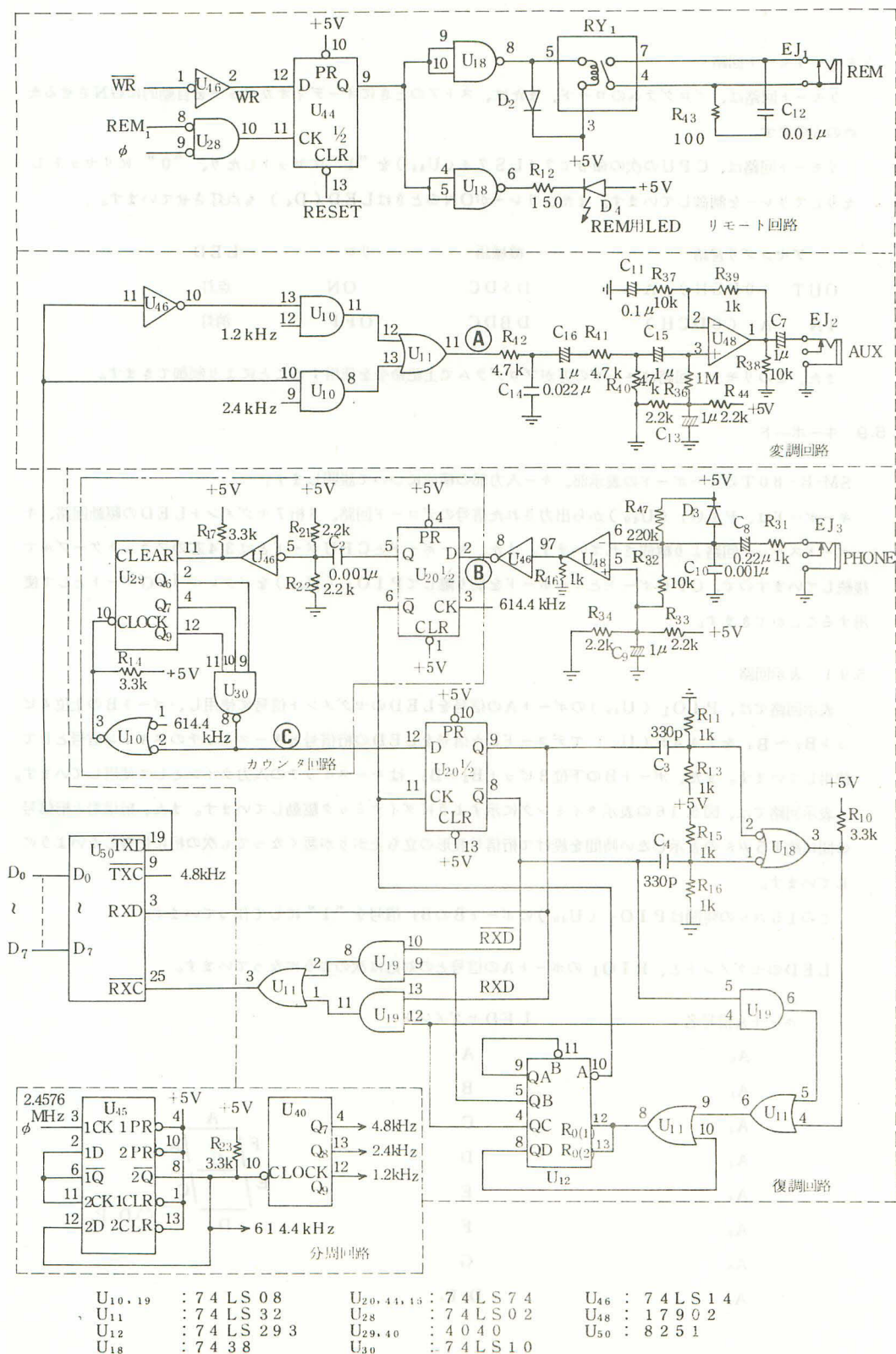



図5.14 オーディオカセットインターフェース回路

5.8.4　リモート回路

リモート回路は、プログラムのロード、または、ストアのときにオーディオカセットを自動的にONさせるためのものです。

リモート回路は、CPUの次の命令で74LS74(U_{44})を"1"にセットしたり、"0"にリセットしたりしてリレーを制御しています。また、リレーがONのときはLED(D_4)も点灯させています。

アセンブリ言語	機械語	リレー	LED
OUT　(0DCH), A	D3DC	ON	点灯
IN　A, (0DCH)	DBDC	OFF	消灯

また、このリモート回路はユーザの方がプログラムで上記命令を使用することにより制御できます。

5.9　キーボード

SM-B-80Tのキーボードの表示部、キー入力部の構成について説明します。

キーボードは、PIO$_1$(U_{49})から出力された信号のデコード回路、8桁7セグメントLEDの駆動回路、キーボードスイッチ回路より構成されています。また、キーボードとCPUボードとは34芯のフラットケーブルで接続していますので、CPUボードとキーボードを切り離してPIO$_1$(U_{49})をパラレルI/Oポートとして使用することができます。

5.9.1　表示回路

表示回路では、PIO$_1$(U_{49})のポートAの信号をLEDのセグメント信号に使用し、ポートBの上位4ビットB_7～B_4を7445(U_5)でデコードした信号をLEDの桁信号とキースイッチのスキャン信号として使用しています。また、ポートBの下位3ビットB_3～B_0はキースイッチの入力ラインとして使用しています。

表示回路では、図5.16の表示タイミングに示すようにダイナミック駆動しています。また、桁信号と桁信号の間に約15μsの表示しない時間を設けて桁信号波形の立ち上がりが悪くなっても次の桁に影響しないようにしています。

この15μsの時間はPIO$_1$(U_{49})のポートBのB_7信号を"1"にして作っています。

LEDのセグメントと、PIO$_1$のポートAの信号との対応は次のようになっています。

ポートA信号名	LEDセグメント
A_0	A
A_1	B
A_2	C
A_3	D
A_4	E
A_5	F
A_6	G
A_7	D.P.

A
F　B
G
E　C
D　○D.P.

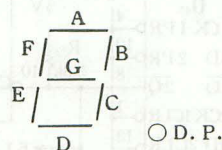
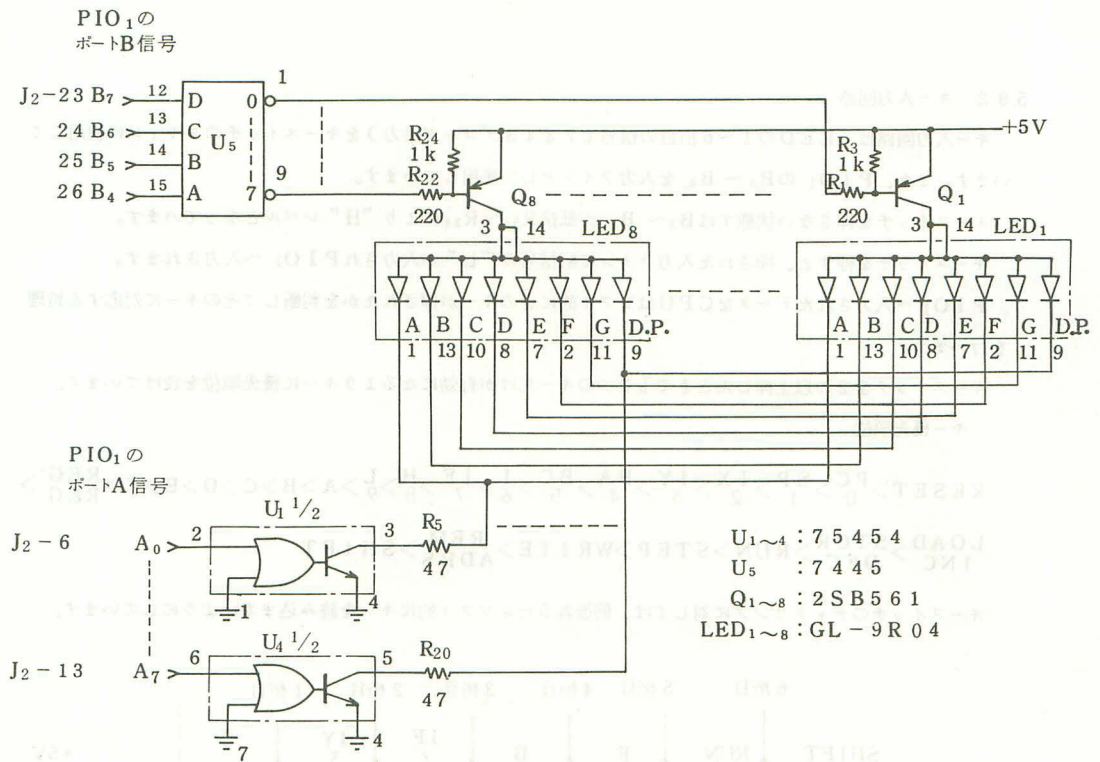



図5.15　表示回路

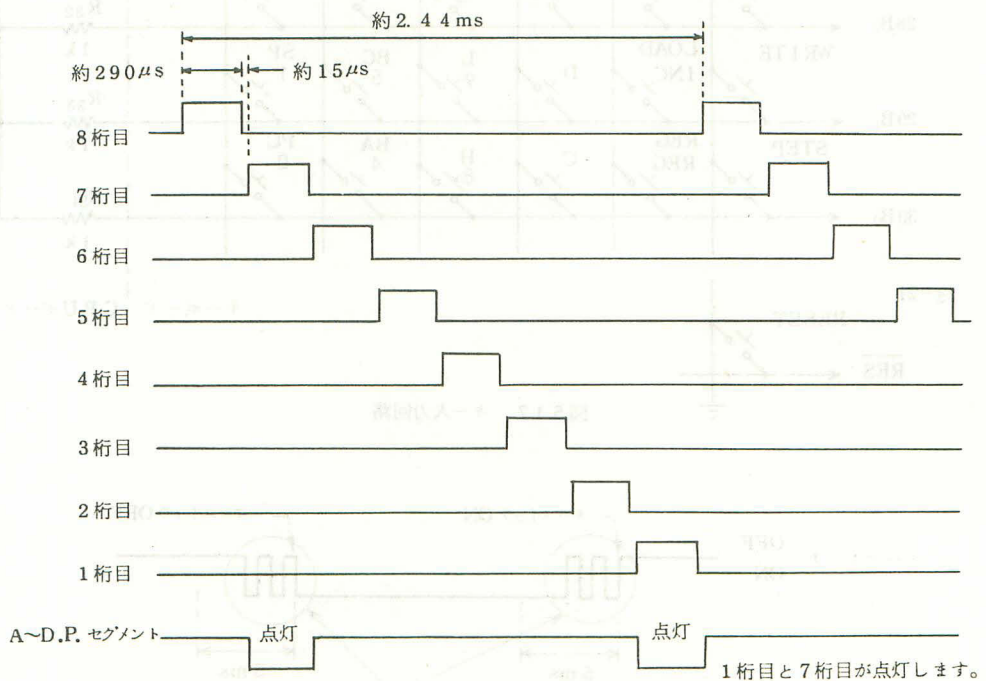



図5.16　表示タイミング波形（LEDへの信号波形）

5.9.2　キー入力回路

キー入力回路は、ＬＥＤの１～６桁目の信号（７４４５デコーダ出力）をキースイッチのスキャンに使用しています。また、ＰＩＯ$_1$ のＢ$_3$～Ｂ$_0$ を入力ラインとして使用しています。

キースイッチを押さない状態ではＢ$_3$～ Ｂ$_0$ が抵抗Ｒ$_{31}$～Ｒ$_{34}$により“Ｈ”レベルとなっています。

キースイッチを押すと、押された入力ラインに桁信号の“Ｌ”が入力されＰＩＯ$_1$ へ入力されます。

ＰＩＯ$_1$ へ入力されたデータをＣＰＵはソフト的にどのキーが押されたかを判断してそのキーに対応する処理を行います。

キースイッチを２つ以上押したときでも１つのキーだけが有効になるようキーに優先順位を設けています。

キー優先順位

ＲＥＳＥＴ＞ＰＣ/0＞ＳＰ/1＞ＩＸ/2＞ＩＹ/3＞ＢＡ/4＞ＢＣ/5＞Ｉ/6＞ＩＦ/7＞Ｈ/8＞Ｌ/9＞Ａ＞Ｂ＞Ｃ＞Ｄ＞Ｅ＞Ｆ＞ＲＥＧ'/ＲＥＧ＞ＬＯＡＤ/ＩＮＣ＞ＳＴＯＲ/ＤＥＣ＞ＲＵＮ＞ＳＴＥＰ＞ＷＲＩＴＥ＞ＲＥＭ/ＡＤＲＳ＞ＳＨＩＦＴ

キースイッチのチャタリングに対しては、前後各５ms ソフト的にキーを読み込まないようにしています。

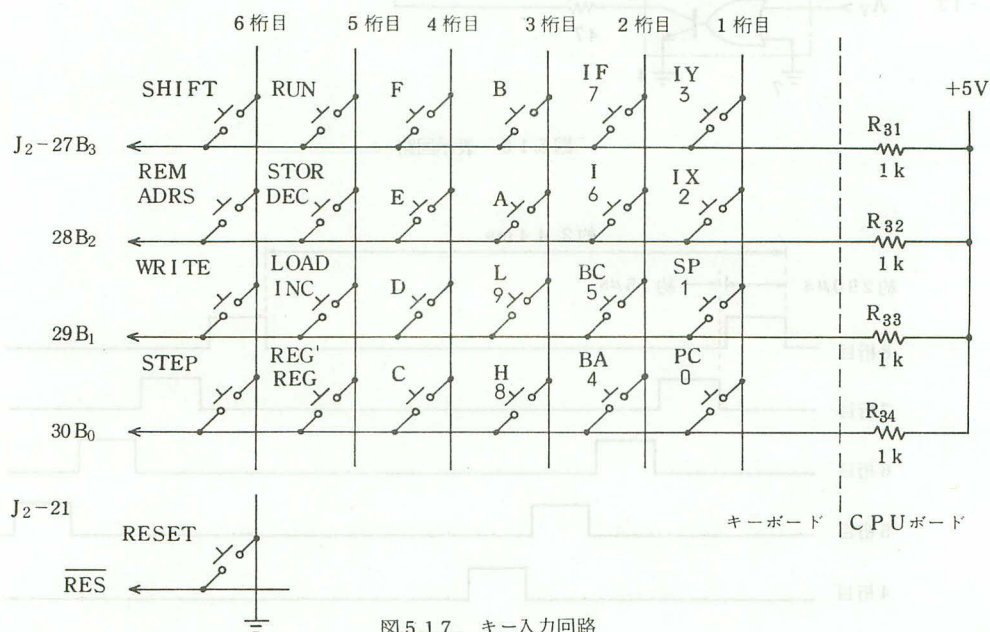



図5.17　キー入力回路

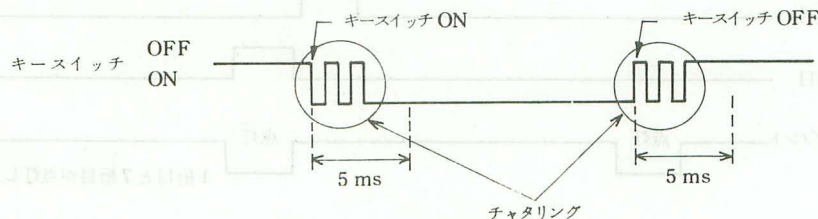



図5.18　キー入力タイミング

5.10　PIO$_1$ 周辺回路

ＣＰＵボードとキーボードとのインターフェースにＰＩＯ$_1$（並列入出力コントローラ）を使用しています。

図5.19 にＰＩＯ$_1$ の周辺回路を示します。

モニタプログラムでは、ＰＩＯ$_1$ のポートＡ８ビットを出力ポートとして、ポートＢのB_0～B_3を入力ポート、B_4～B_7 を出力ポートとして使用しています。

ＰＩＯ$_1$ はモニタプログラムでは割り込みが掛からないようにして使用していますが、キーボードを使用せずに他の周辺装置の接続を考慮し、割り込み優先回路を設けています。

割り込み優先は、ボード内ではＰＩＯ$_2$ より上位に設定しています。

ポートＢのB_7 ラインはＲＥＳＥＴ信号とＯＲＧＡＴＥを構成していますが、これは、システムをリセットしたときにＬＥＤ（７セグメントの）を全桁消灯するための回路です。このためポートＢを他の目的に使用するときはジャンパ線 P_2 と P_3 を切断し、P_1 に新たにジャンパ線を追加すれば J_2（３４ピン）にＰＩＯ$_1$ の B_7 を直接取り出せます。

U_{21} ：７４ＬＳ３２
U_{37} ：７４ＬＳ０８
U_{49} ：ＬＨ－００８１

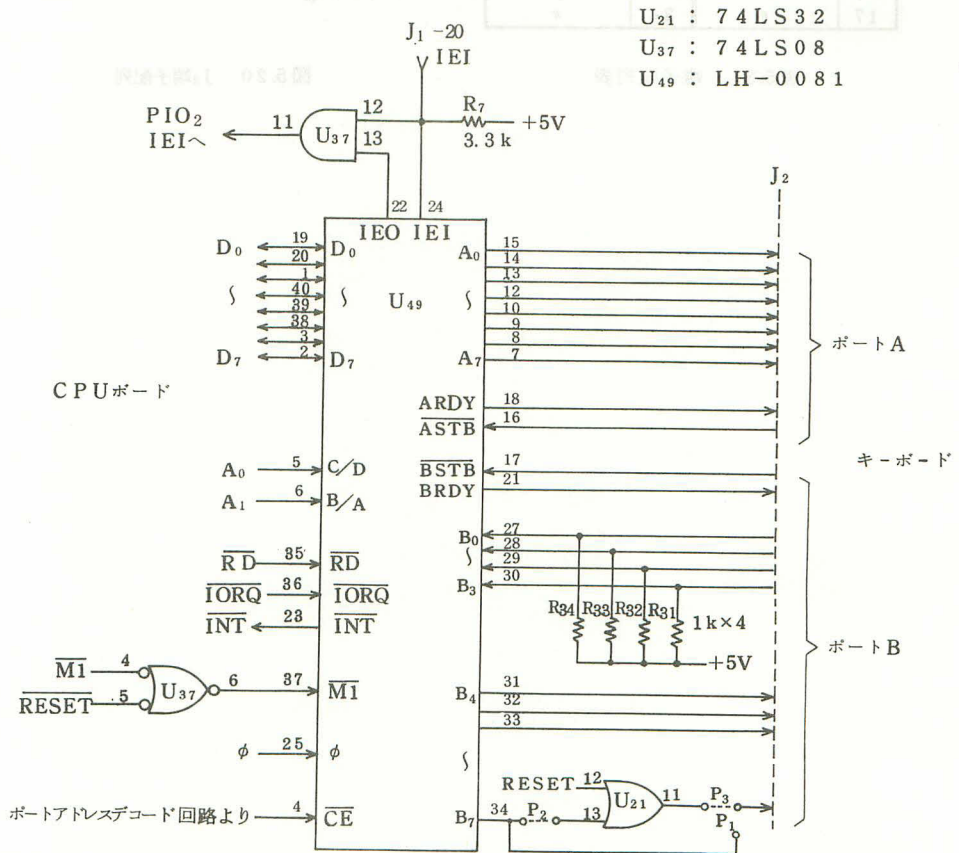



図5.19　ＰＩＯ$_1$ 周辺回路

端子 №.	信号名	端子 №.	信号名
1	V_{CC}(+5V)	18	V_{CC}(+5V)
2	〃	19	〃
3		20	
4		21	$\overline{RES}$
5	$\overline{PS_1}$	22	$\overline{PS_2}$
6	A_7	23	B_7
7	A_6	24	B_6
8	A_5	25	B_5
9	A_4	26	B_4
10	A_3	27	B_3
11	A_2	28	B_2
12	A_1	29	B_1
13	A_0	30	B_0
14	$\overline{ASTB}$	31	$\overline{BSTB}$
15	ARDY	32	BRDY
16	GND	33	GND
17	〃	34	〃

表5.3　端子配列表

信号の意味は付録6の端子信号説明表を参照してください。

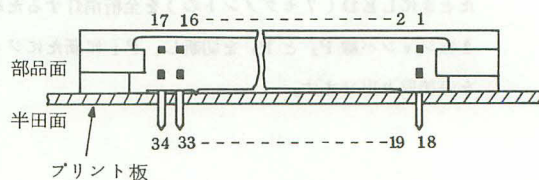
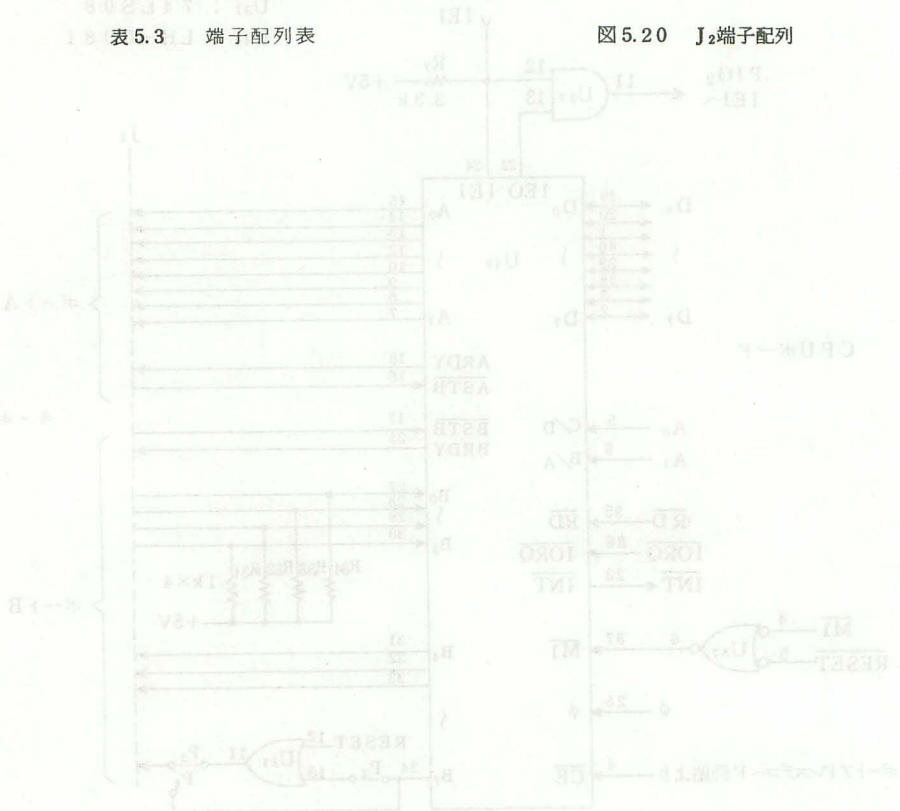



図5.20　J_2端子配列

第6章　システムの拡張

SM-B-80Tでは、ユーザ用としてI/Oポート（PIO2）、バスドライバなどを拡張できますが、その拡張方法、使用方法を説明します。

6.1　1ビット出力ポート（74LS74―U_{44}）

CPUボードにポートアドレスDDを割り当てたD型F/Fを設けています。

このF/Fはプログラムで制御できますのでリレーの制御や、音楽の自動演奏に使用できます。

6.1.1　ハードウェア

D型F/F（U_{44}）はCPUの出力命令で"H"にセットし、入力命令で"L"にリセットできる回路構成となっています。

74LS74（U_{44}）のQ，$\bar{Q}$出力はCPUボード上にCHECKピンを設けていますのでここに半田付、または、ワイヤーラピングを行って外部へ信号を取り出せます。

システムのリセット後、74LS74（U_{44}）は"L"にリセットされます。

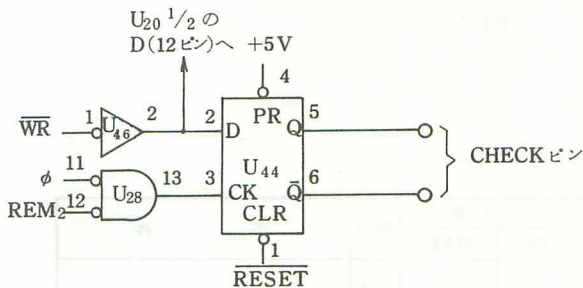
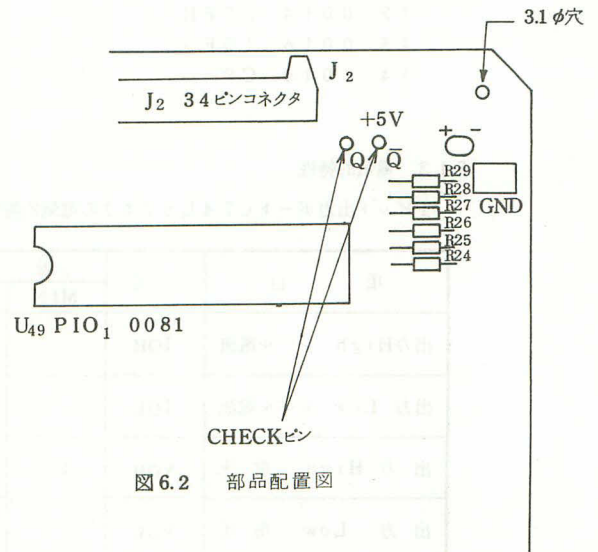



図6.1　U_{44}　回路図　　　図6.2　部品配置図

6.1.2　使用方法

U_{44}のD型F/Fは次のCPUの命令でセット、リセットできます。

アセンブリ言語	機械語	U_{44} Q	$\bar{Q}$
OUT　(0DDH), A	D3DD	1（"H"）	0（"L"）
IN　A, (0DDH)	DBDD	0	1

（例）

3.4.1項でリモート用LEDを約1秒周期で点滅させましたが、同じ方法で74LS74（U_{44}）を"H"にセット、"L"にリセットを繰り返すプログラムを示します。

このプログラムは、3.4.1項のプログラムのポートアドレスをDCからDDに変更するだけで作成できます。

```
                Z-80  ASSEMBLER  V1. 1  PAGE   1

      アドレス  機械語                アセンブリ言語
  1                                ORG    0
  2   0000   D3DD      LOOP:       OUT    (0DDH),A
  3   0002   CD0C00                CALL   TCOUNT
  4   0005   DBDD                  IN     A,(0DDH)
  5   0007   CD0C00                CALL   TCOUNT
  6   000A   18F4                  JR     LOOP
  7   000C   0605      TCOUNT:     LD     B,5
  8   000E   210024    LOOP1:      LD     HL,2400H
  9   0011   2B        LOOP2:      DEC    HL
 10   0012   7C                    LD     A,H
 11   0013   B5                    OR     L
 12   0014   20FB                  JR     NZ,LOOP2
 13   0016   10F6                  DJNZ   LOOP1
 14   0018   C9                    RET
```

6.1.3　電気的特性

1ビット出力ポート（74LS74）の電気的特性を示します。

項　　目	記号	規	格		単位	条　　件
		MIN	TYP	MAX		
出力High　レベル電流	I_{OH}			−400	μA	
出力 Low　レベル電流	I_{OL}			8	mA	
出 力 High　電 圧	V_{OH}	2.7	3.4		V	V_{CC}= 4.75 V　V_{IH}= 2 V V_{IL}= 0.8 V　I_{OH}= −400μA
出 力　Low　電 圧	V_{OL}		0.35	0.5	V	V_{CC}= 4.75V　I_{OL}= 8 mA V_{IL} = 0.8 V

※　74LS74はT・I製を使用していますので特性の詳細は、T・Iデータブックを参照してください。

表6.1　74LS74電気的特性

6.2　パラレルI／Oポート（PIO_2）

CPUボード内では、ユーザ用 I／Oポートとして PIO　LH−0081（8ビット×2ポート）を1個増設できます。またPIOの I／OラインはJ_3（44ピン）へ出力しています。

6.2.1　ＰＩＯ２周辺回路

ＰＩＯ２は、ポートアドレスＤ４～Ｄ７の４ポートを割り当てています。

ＰＩＯ２は、システムのリセット後モニタプログラムではモード設定など行っていません。このためＰＩＯ２を使用するときは使用目的に応じてプログラムでモードなどの設定をしてください。また、ＰＩＯ２の割り込みに対する優先順位はＰＩＯ１より下位に設定しています。

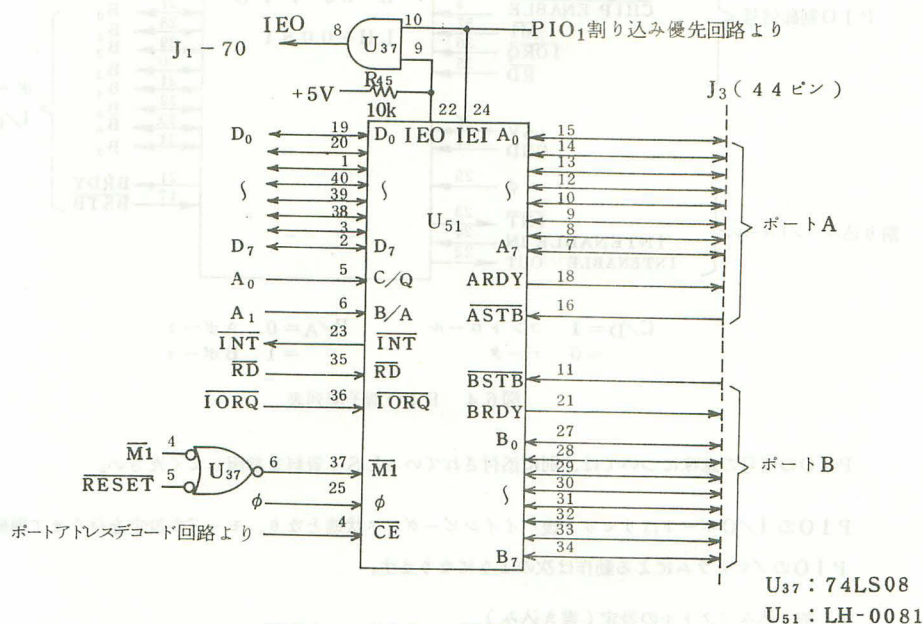



図6.3　　ＰＩＯ２　周辺回路

Ｊ２（４４ピン）の端子配列は付録５の端子配列表を参照してください。

6.2.2　ＰＩＯのプログラミング法

ＰＩＯはプログラマブルな並列入出力コントローラで、プログラムにより２つのＩ／Ｏポートをモード0, 1, 2, 3 のいずれかに指定して使用できます。

ＰＩＯは、紙テープパンチャ、紙テープリーダ、プリンタ、キーボードなどの周辺装置とＺ－８０　ＣＰＵをＴＴＬレベルでインターフェースするものです。

図6.4にＰＩＯの端子配列を示します。また、ポートアドレスの詳細は5.3.2項を参照してください。

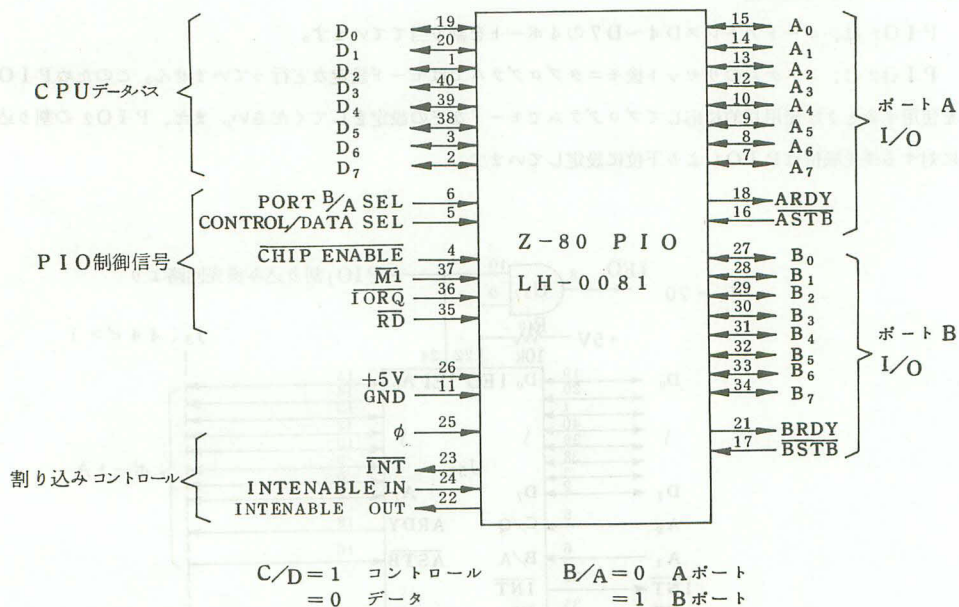



図6.4　PIO端子配列表

PIOの信号の意味については、別に添付されているLSI資料を参照してください。

PIOのI/Oポートはリセット後ハイインピーダンス状態となり、モードの設定を行うまで継続します。

PIOのプログラムによる動作は次のようになります。

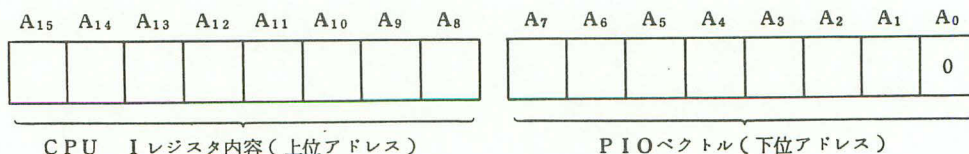
(1) 割り込みベクトルの設定（書き込み）

PIOは、CPUと一緒に使ってモード2の割り込みに使用できます。モード2では、割り込みをかけているデバイスが割り込みベクトルをデータバスに乗せ、CPUはこのベクトルを割り込み処理ルーチンの下位アドレスとします。また、上位アドレスはあらかじめIレジスタに書き込んでおきます。

PIOにベクトルを書き込むときは次の形式のコントロール語を希望するポートに書き込みます。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
V_7	V_6	V_5	V_4	V_3	V_2	V_1	0

この0が割り込みベクトルを表わします。

割り込みを処理するルーチンのアドレス（16ビット）は次のようになります。

（例）

割り込み処理ルーチンの開始アドレスが書き込まれているメモリのアドレスをD124番地とし、PIO2のポートAにベクトル24を設定します。また、CPUのIレジスタには上位アドレスのD1を設定しておきます。

○CPUのIレジスタにD1を設定します。

アセンブリ言語		機械語	
LD	A，0D1H	3ED1……	AレジスタにD1をロード
LD	I，A	ED47……	IレジスタにAレジスタの内容をロード

○PIO2のポートAにベクトル24を設定します。

LD	A，24H	3E24……	Aレジスタに24をロード
OUT	（0D5H），A	D3D5……	ベクトルを設定

↑ PIO2のポートアドレス（0D5H）

(2) モードの設定

PIOにはモード0, 1, 2, 3 の4つのモードがあります。

モードの設定は次のように行います。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
M_1	M_0	×	×	1	1	1	1

×＝未使用ビット（0でも1でも可）

D_7〜D_6：モード語を表わします。　D_3〜D_0：モード設定を表わします。

モード		M_1	M_0
出力モード	0	0	0
入力モード	1	0	1
双方向性モード	2	1	0
ビットコントロールモード	3	1	1

モード0 ： データを周辺装置へ出力します。

モード1 ： データを周辺装置から入力します。

モード2 ： 周辺装置からデータの入力、周辺装置へのデータの出力ができます。……ポートAのみ。

モード3 ： ポートをビット単位に入力／出力に指定して使用できます。

（例）

PIO2のポートAを出力モード（モード0）に、ポートBを入力（モード1）に指定して、ポートAに10101010（AA）を出力します。

oポートAを出力モードに設定します。

アセンブリ言語		標準語	
LD	A，0FH	3E0F……	AレジスタにOFをロード
OUT	(0D5H)，A	D3D5……	モード0を設定

oポートBを入力モードに設定します。

LD	A，4FH	3E4F……	Aレジスタに4Fをロード
OUT	(0D7H)	D3D7……	モード1を設定

oポートAに10101010(AA)を出力します。

LD	A，0AAH	3EAA……	AレジスタにAAをロード
OUT	(0D4H)，A	D3D4……	ポートAにAレジスタ内容を出力

モード3を設定したときは、次のコントロール語を設定してポート(8ビット)のどのビットを入力にするか、出力にするかを指定します。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O

I/O＝1　ビットを入力用に指定します。
　　＝0　ビットを出力用に指定します。

(3) 割り込みコントロール語の設定

ポートを割り込みに使用するときは、次に示す割り込みコントロール語を設定します。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
割り込み許可フラグ	AND/OR	High/LOW	MASK follow	0	1	1	1

D_6～D_4：モード3でのみ使用します。　D_3～D_0：割り込みコントロール語を表わします。

D_7 ＝1　割り込み許可フラグをセットし、CPUへ割り込み要求を発生できます。
　　＝0　割り込み許可フラグをリセットし、CPUへ割り込み要求を発生できません。

D_6 ＝1　AND　マスクされていないビットのすべてがD_5で指定される状態になったときに割り込みが発生します。
　　＝0　OR　マスクされていないビットのどれかがD_5で指定される状態になったときに割り込みが発生します。

D_5 ＝1　High　ポートデータバスラインが"H"になると割り込みを発生します。
　　＝0　Low　ポートデータバスラインが"L"になると割り込みを発生します。

D_4　D_4が1であれば次にポートに書かれるコントロール語は下記のようなマスクとして取り扱われます。

MB(マスクビット)＝0であるビットだけが割り込み発生を監視されます。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
MB_7	MB_6	MB_5	MB_4	MB_3	MB_2	MB_1	MB_0

割り込み許可フラグは次に示すコントロール語でセット、リセットできます。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
割り込み許可フラグ	×	×	×	0	0	1	1

このコントロール語は、割り込み許可フラグだけをセットしたり、リセットしたりして他のビットはそのままにしておきたいときに使用します。

（例）

PIO2のポートBをモード3（ビットコントロールモード）で使用し、B_0 ～ B_3 を入力、B_4 ～ B_7 を出力にします。また、割り込みについてはB_0 ～ B_1 の2ビットだけを監視し、B_0、または、B_1 のどちらかが"H"になったときに割り込みが発生するようにします。

割り込み処理ルーチンの開始アドレスが書き込まれているメモリのアドレスをD124番地とします。

○CPUのIレジスタにD1を設定します。

アセンブリ言語	機械語
LD　A，0D1H	3ED1
LD　I，A	ED47

Iレジスタ

1	1	0	1	0	0	0	1

○PIO2のポートBにベクトルを設定します。

LD　A，24H	3E24
OUT　(0D7H)，A	D3D7

割り込みベクトル

0	0	1	0	0	1	0	0

○モード　を設定します。

LD　A，0CFH	3ECF
OUT　(0D7H)，A	D3D7

モード設定

1	1	×	×	1	1	1	1

○モード3を設定したので I／O指定をします。

LD　A，0FH	3E0F
OUT　(0D7H)，A	D3D7

I／O指定

0	0	0	0	1	1	1	1

○割り込みコントロール語を設定します。

LD　A，0B7H	3EB7
OUT　(0D7H)，A	D3D7

割り込みコントロール語

1	0	1	1	0	1	1	1

○割り込みのマスクを設定します。

LD　A，0FCH	3EFC
OUT　(0D7H)，A	D3D7

マスクの設定

1	1	1	1	1	1	0	0

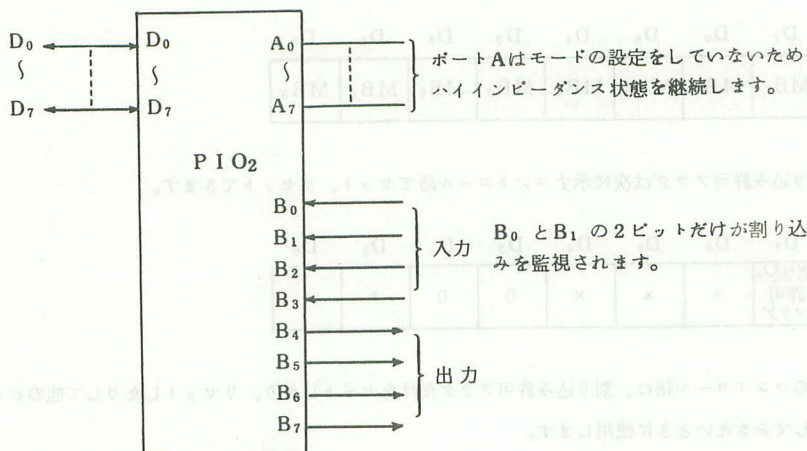



図6.5　PIO_2の設定例

6.3　キーボードインターフェース（PIO_1）

キーボードのインターフェースとして使用しているPIO_1については5.10項で説明しましたが、PIO_1を他の目的に単独で使用する場合について説明します。

PIO_1は、モニタプログラムではキーボードとのインターフェース用にモード設定をしていますので単独に使用するときは改めてプログラムで設定し直す必要があります。

プログラムの方法はPIO_2で説明した方法で行いますが、ポートアドレスが次のようにPIO_2と異なります。

	ポートアドレス	
	PIO_1	PIO_2
ポートAデータ	D0	D4
ポートAコントロール	D1	D5
ポートBデータ	D2	D6
ポートBコントロール	D3	D7

表6.2　PIO_1，PIO_2ポートアドレス

キーボードを使用せずにPIO_1を他の目的に使用する場合は、図6.6に示すようにプリント板に取り付けているP_2とP_3のジャンパ線を切断し、新たにP_1の位置にジャンパ線を取り付けてください。

これによりJ_2（34ピン）へはPIO_1のB_7信号が直接つながります。

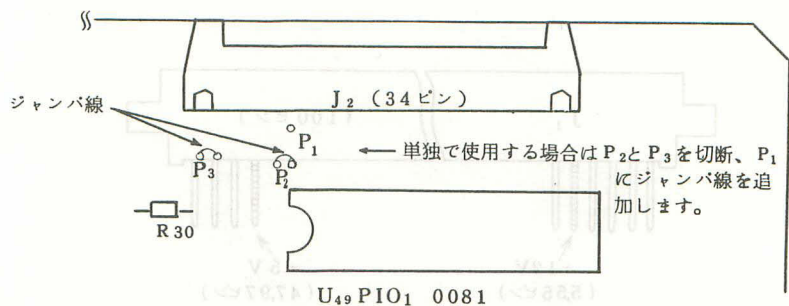



図6.6 ジャンパ線

J_2（34ピン）の端子配列は、5.10項、または、付録6を参照してください。

6.4 EPROM使用方法

SM-B-80TではROMとしてモニタプログラムを書き込んだバイポーラPROM LH-7055を1個実装し、ユーザ用としてさらに1個増設できますが、このバイポーラタイプ以外に2708タイプのEPROMを使用できます。

2708タイプのEPROMを使用する場合は＋12Vと－5Vの電源が＋5V以外に必要です。

2708タイプのEPROMを使用する方法について説明します。

＋12Vと－5VをSM-B-80Tに接続するには、図6.8に示す＋12Vと－5Vの配線部分に半田付けするか、図6.7に示すJ_1（100ピン）の＋12Vと－5V用端子に半田付けします。

	J_1 端子NO	
＋12V	5ピン	55ピン
－5V	47ピン	97ピン

また、電源ノイズによる誤動作を防ぐため、図6.8に示すプリント板のバックシンボルの位置にコンデンサを取り付けてください。

コンデンサには次のものを使用します。

コンデンサNO	部品
C_A～C_F	25V以上の1 μF タンタルコンデンサ
$C_{17,\ 18}$	25V以上の33 μF電解コンデンサ

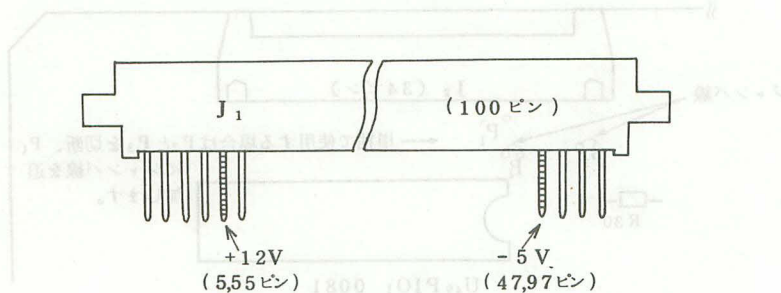
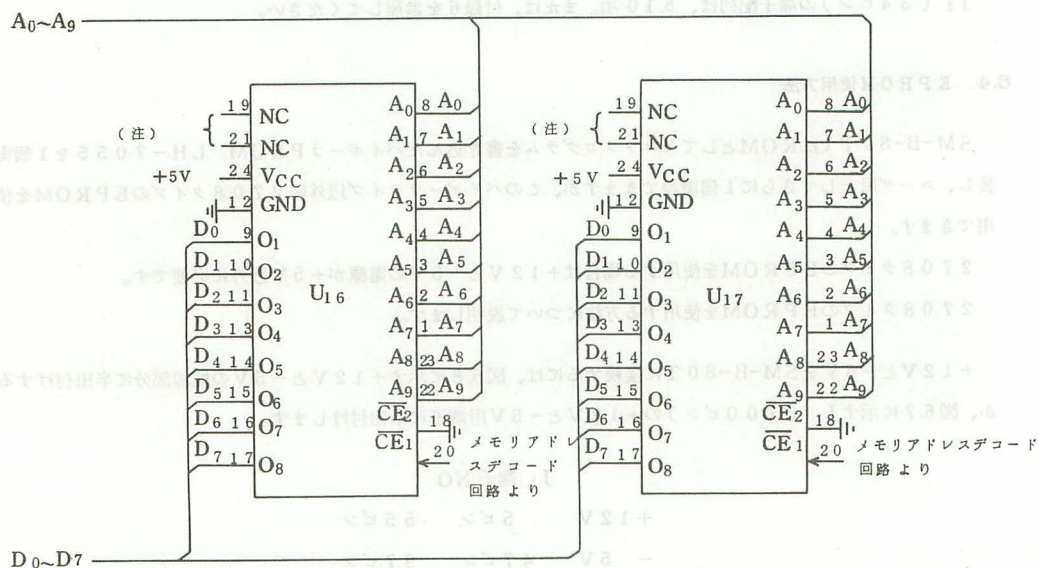



図 6.7　J₁（100ピン）への電源配線

U₁₆,₁₇ ： LH－7055または2708タイプEPROM

（注）2708タイプEPROMを使用する場合、19ピンに＋12V、21ピンに－5Vが供給されます。

図 6.9　ROM回路

6.5　コントロール信号用バッファ

コントロール信号用バッファ回路は、8T97（U₃）、8T28（U₄）、7404（U₉）で構成します。

コントロール信号の中でBUSRQ* はダイレクトメモリアクセス（DMA）に使用します。また、MRESET* は、ボードの外部からシステムをリセットするための入力信号ラインです。

2708使用時
電源配線+12V，-5V
2708使用時追加
REM
AUX
PHONE
E000 — 0000
EJ1
EJ2
EJ3
ADRS SW1
+5V
GND
J2
J3
U49 PI01 0081
U50 8251
U51 PI02 0081
RY1
XTAL
U42 CPU 0080
ϕCK
U16 7055
U17 7055
U24 2114
U25 2114
U26 2114
U33 2114
U34 2114
U35 2114
U23 2111
U32 2111
U1 T26
U2 T26
U3 T97
U4 T28
U5 T26
U6 T26
U7 T26
U8 T26
U9 04
-5V
J1-50
図 6.8 シンボル図面

6.6 双方向性アドレスバッファ

双方向性アドレスバッファ回路は、8T26を4個(U_5 ～ U_8) 使用して構成します。

アドレスバッファは通常ボードの外部へアドレス信号を出力していますが、ボード内のメモリをアクセスするためにBUSRQ* を"L"にしてDMA (ダイレクトメモリアクセス)を使用するときに、アクセスしたいメモリのアドレスを入力できます。

BUSRQをCPUが受け付けるとCPUのアドレスラインはフローティングになります。また、BUSAK* も"L"となり、8T26はレシーブ状態になります。

双方向性アドレスバッファは、アドレス信号を負論理で出力しています。また、バッファ用ICの取り付け位置などは、2.5項を参照してください。

6.7 双方向性データバッファ

双方向性データバッファは、8T26を2個(U_1～U_2)使用して構成します。

図6.10 に回路を示します。

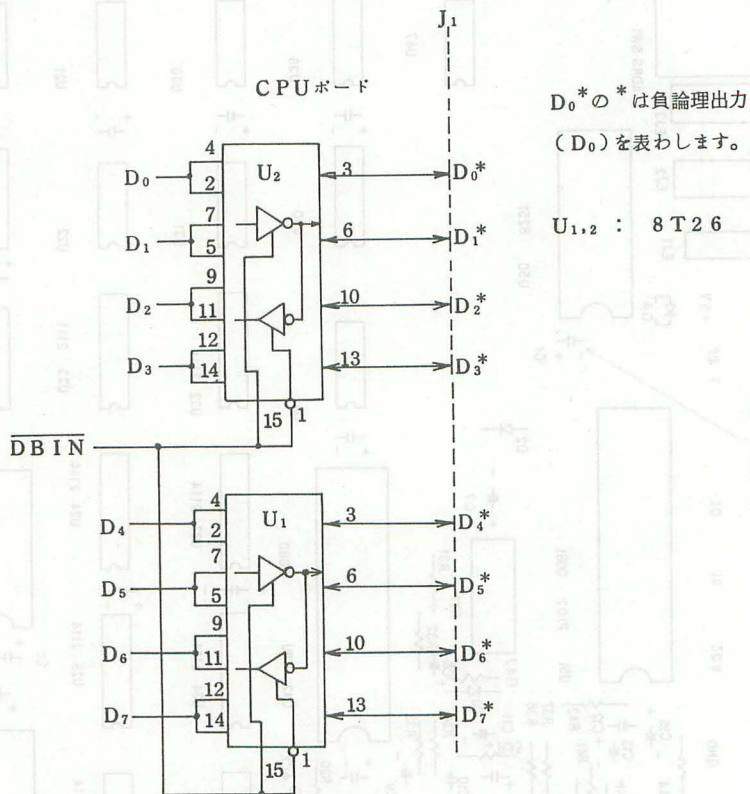



図6.10 双方向性データバッファ回路

SM-B-80Tでは、双方向性データバッファを制御する回路をボード内に含んでいませんので、ボードの外でメモリなどを使用するときは制御回路を追加してください。

図6.11に制御回路例を示します。

(例)

ボードの外に8Kバイトのメモリ(スタティックRAM)を追加します。メモリは、アドレスC000番地からの8Kバイトに置いています。

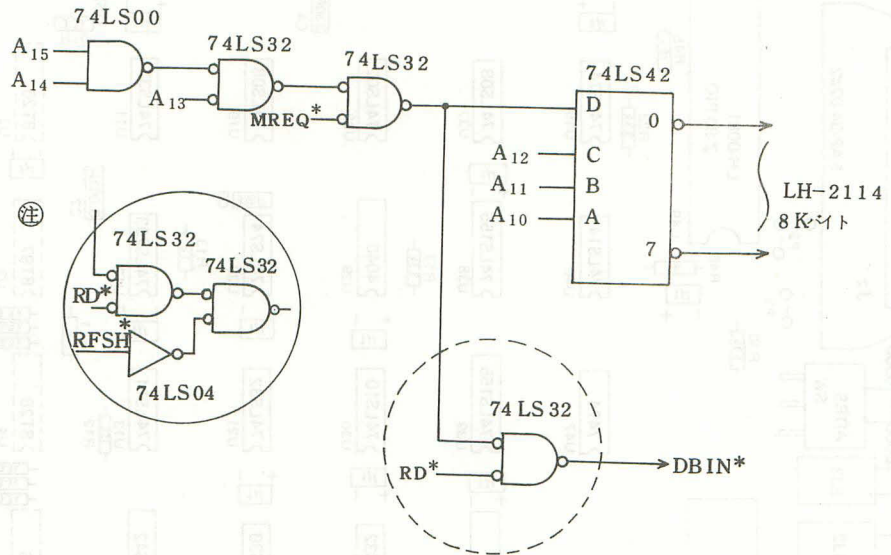



図6.11　双方向性データバッファ制御回路例

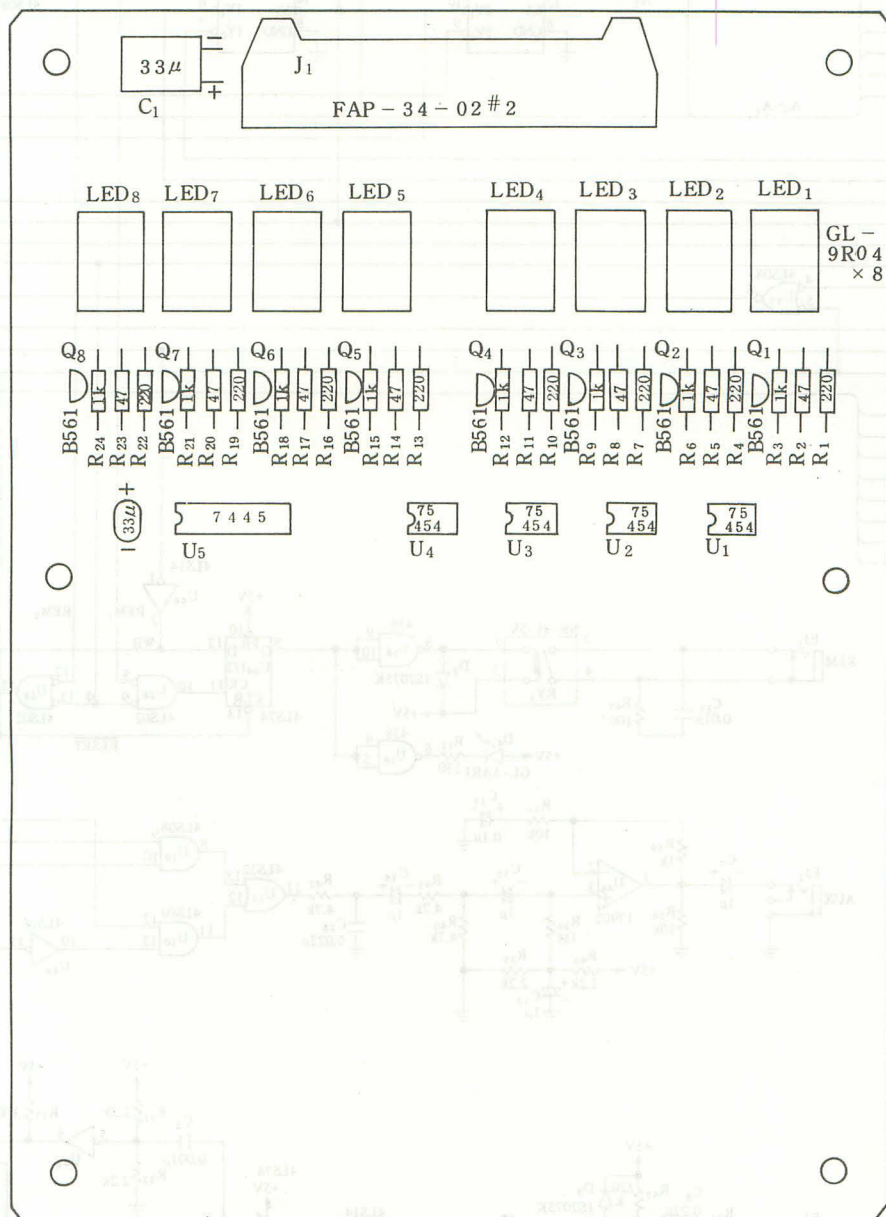
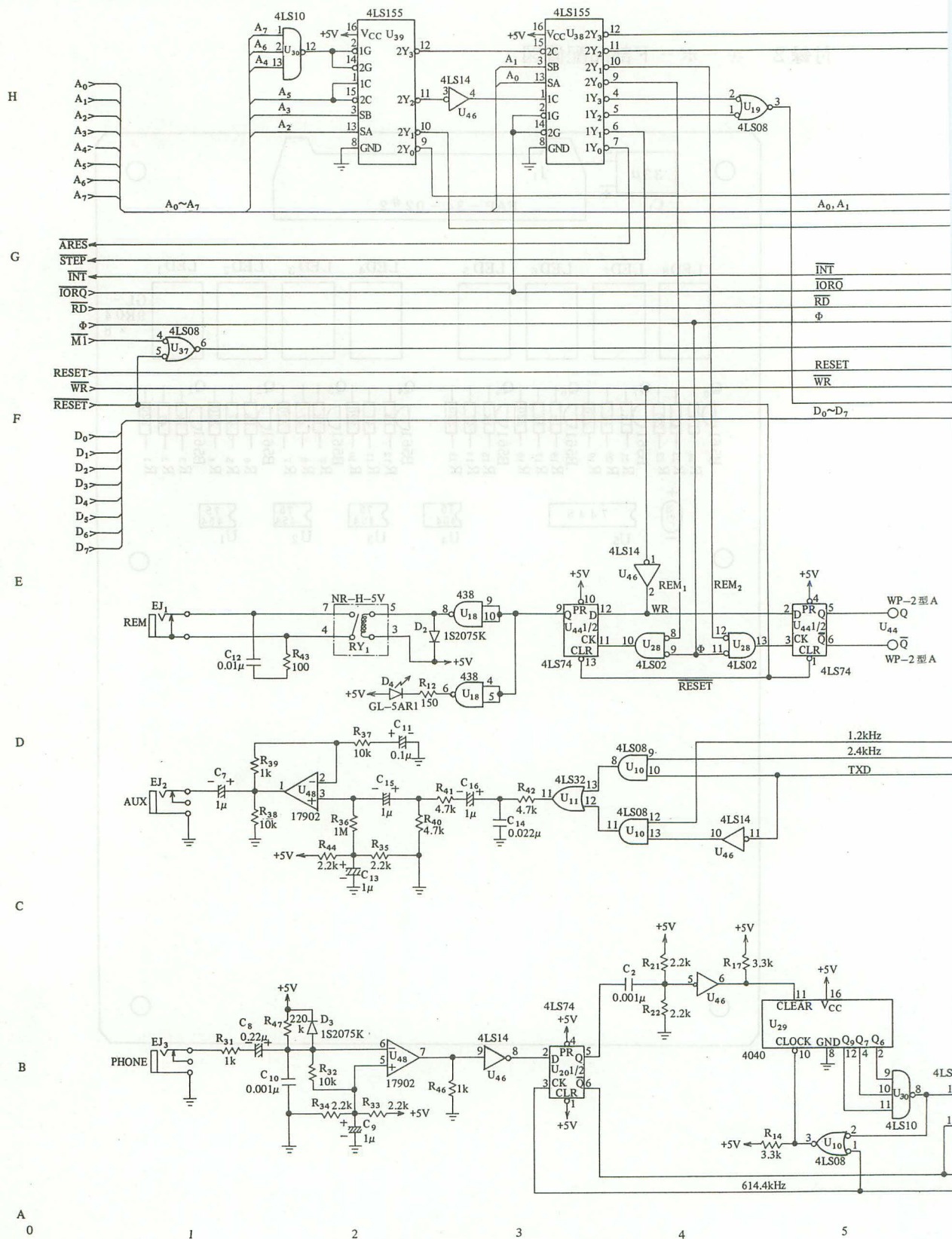
(注) ボードの外にダイナミックメモリを使用する場合は 注 の回路を使用します。

双方向性アドレスバッファ、データバッファに使用する8T26の代わりに8T28を使用しますと、アドレス信号、データ信号を正論理で出力することができます。

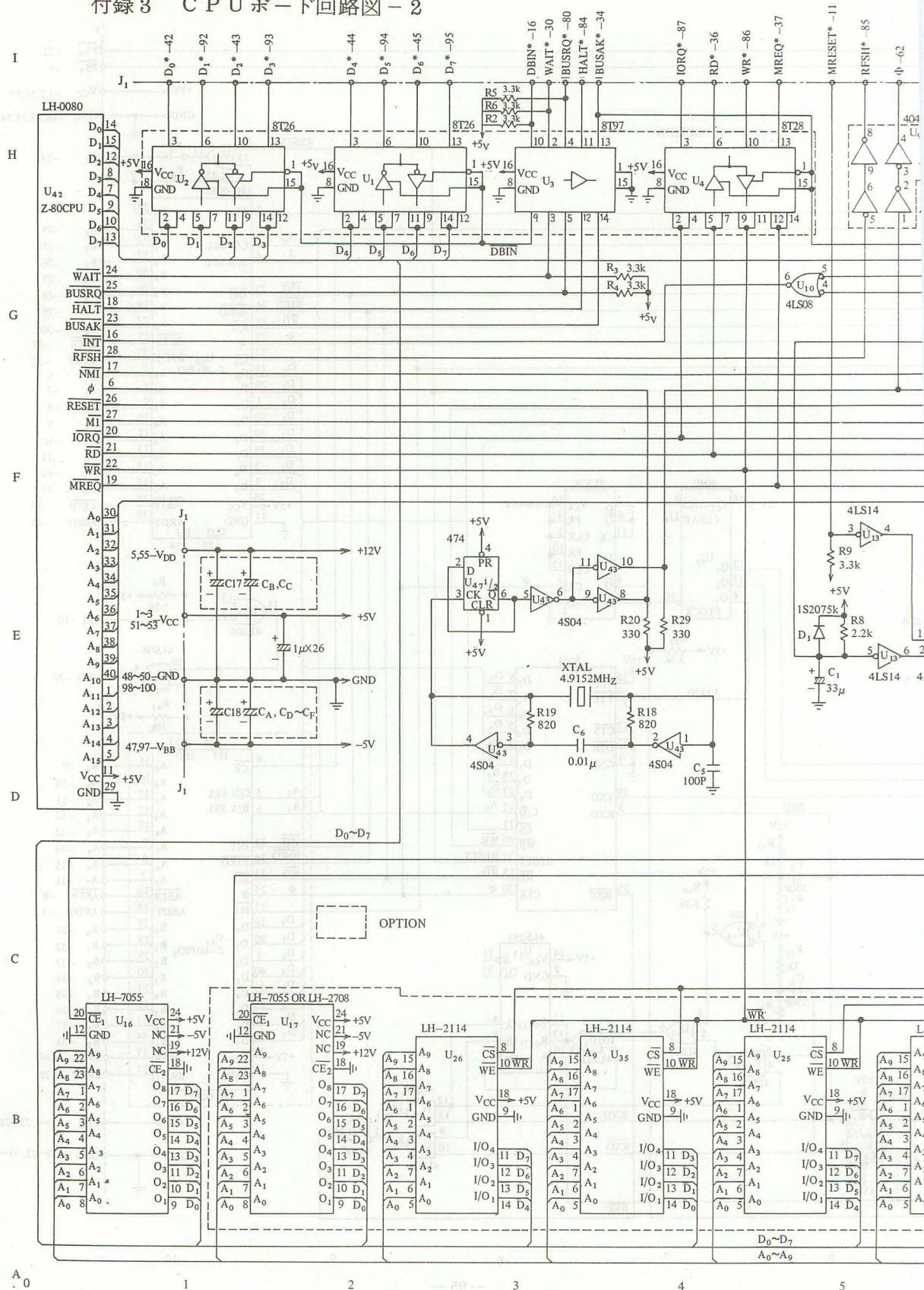
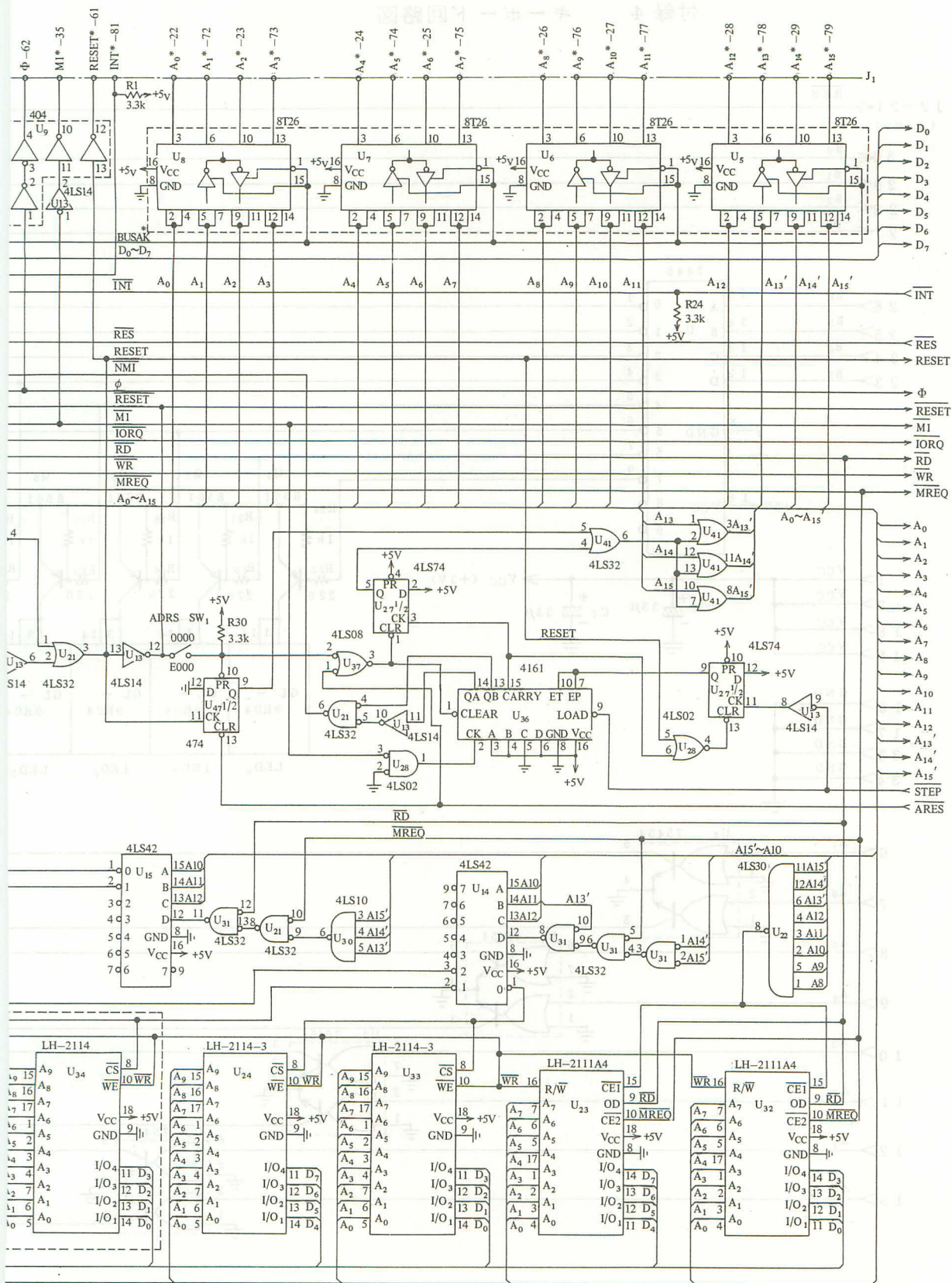
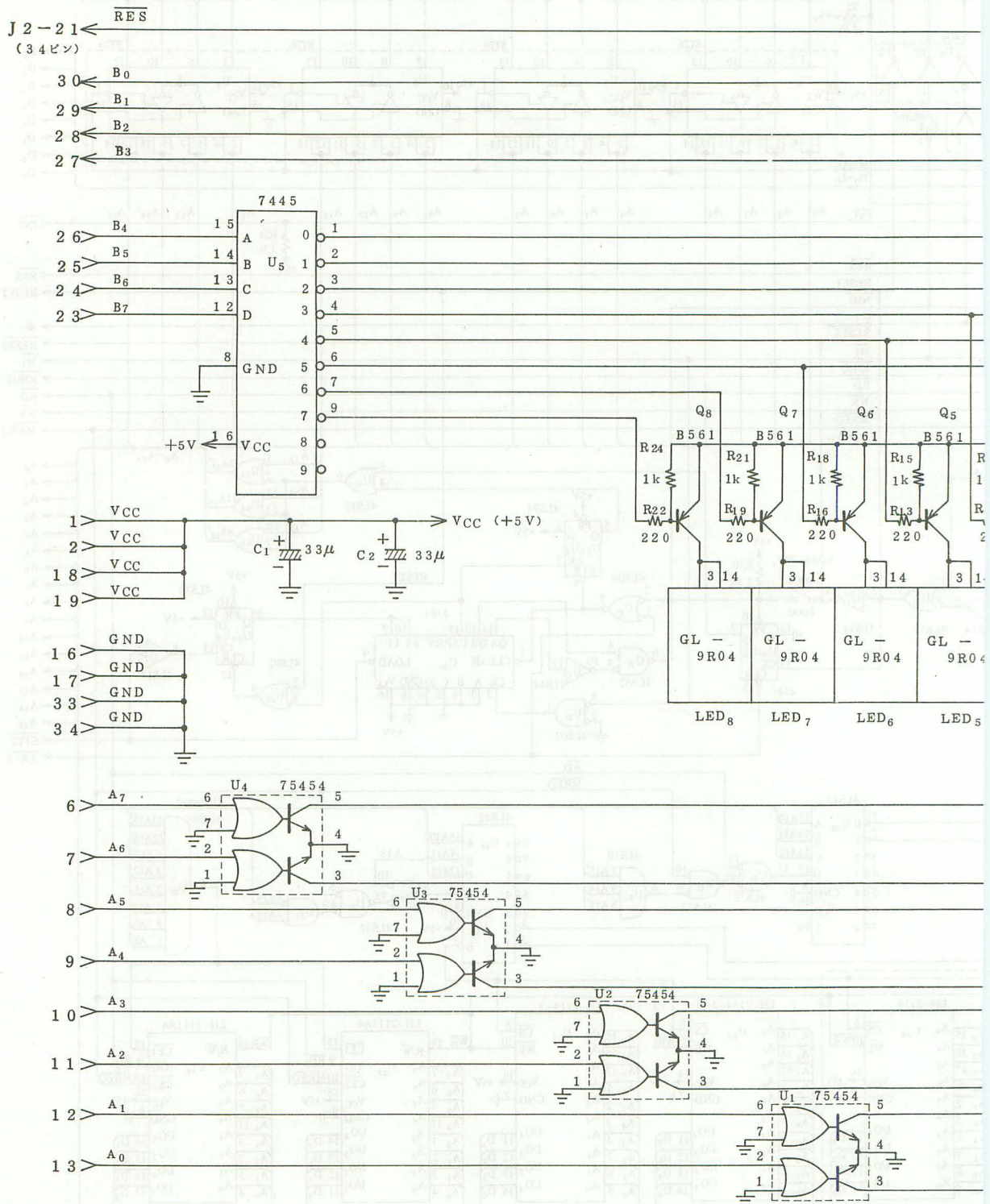
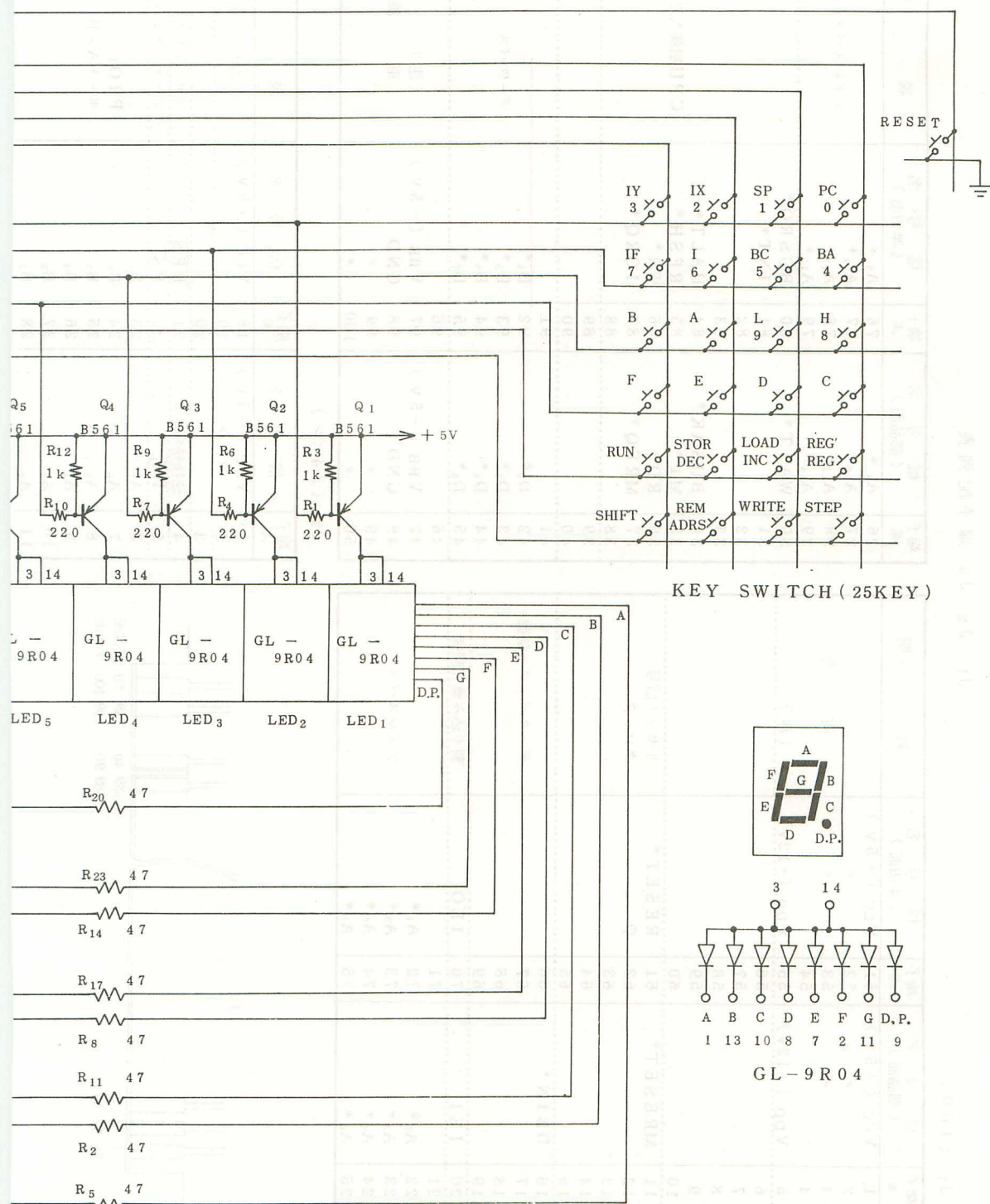
付録1　ＣＰＵボード部品配置図

付録２　キーボード部品配置図

付録3　CPUボード回路図－1

OPTION
RESET
4LS32
LH-0081
Z-80 PIO1
4040
4LS74
8251
4.8kHz
RXD
TXD
4LS08
4LS293
Z-80PIO2
+5V
GND

付録3　CPUボード回路図－2

付録 4　キーボード回路図

RESET
IY 3
IX 2
SP 1
PC 0
IF 7
I 6
BC 5
BA 4
B
A
L 9
H 8
F
E
D
C
RUN
STOR DEC
LOAD INC
REG' REG
SHIFT
REM ADRS
WRITE
STEP
KEY SWITCH (25KEY)
Q5
Q4
Q3
Q2
Q1
B561
+5V
R12 1k
R9 1k
R6 1k
R3 1k
R10 220
R7 220
R4 220
R1 220
3 14
GL - 9R04
LED5
LED4
LED3
LED2
LED1
A
B
C
D
E
F
G
D.P.
R20 47
R23 47
R14 47
R17 47
R8 47
R11 47
R2 47
R5 47
GL-9R04

LH8HO2-KEY

付録5　端子配列表

J1　J2　J3 端子配列表

J1 （100ピン）

端子No.	信号名（部品面）	端子No.	信号名（半田面）	説明
1	VCC（+5V）	51	VCC（+5V）	
2	〃	52	〃	
3	〃	53	〃	電源
4		54		
5	VDD（+12V）	55	VDD（+12V）	（注）
6		56		
7		57		
8		58		
9		59		
10		60		
11	MRESET*	61	RESET*	リセット信号
12		62	φ	クロック
13		63		
14		64		
15		65		
16	DBIN*	66		
17		67		データバッファ制御
18		68		
19		69		
20	IEI	70	IEO	割り込み優先制御
21		71		
22	A_0*	72	A_1*	
23	A_2*	73	A_3*	アドレスバス
24	A_4*	74	A_5*	
25	A_6*	75	A_7*	

端子No.	信号名（部品面）	端子No.	信号名（半田面）	説明
26	A_8*	76	A_9*	
27	A_{10}*	77	A_{11}*	アドレスバス
28	A_{12}*	78	A_{13}*	
29	A_{14}*	79	A_{15}*	
30	WAIT*	80	BUSRQ*	
31		81	INT*	
32		82		
33		83		
34	BUSAK*	84	HALT*	
35	M1*	85	RFSH*	CPU制御入力
36	RD*	86	WR*	
37	MREQ*	87	IORQ*	
38		88		
39		89		
40		90		
41		91		
42	D_0*	92	D_1*	
43	D_2*	93	D_3*	データバス
44	D_4*	94	D_5*	
45	D_6*	95	D_7*	
46		96		
47	VBB（−5V）	97	VBB（−5V）	（注）
48	GND	98	GND	電源
49	〃	99	〃	
50	〃	100	〃	

J2 （34ピン）

端子No.	信号名	端子No.	信号名	説明
1	VCC（+5V）	18	VCC（+5V）	電源
2	〃	19	〃	
3		20		
4	逆接続防止ピン	21	$\overline{\mathrm{RES}}$	リセット入力
5	$\overline{\mathrm{PS_1}}$	22	$\overline{\mathrm{PS_2}}$	ポートアドレス出力
6	A_7	23	B_7	
7	A_6	24	B_6	PIO1
8	A_5	25	B_5	ポートA，Bバス
9	A_4	26	B_4	
10	A_3	27	B_3	
11	A_2	28	B_2	

J1

1 2 3　39 40　49 50　部品面
51 52 53　89 90　99 100　半田面

12	A_1	29	B_1	
13	A_0	30	B_0	
14	$\overline{ASTB}$	31	$\overline{BSTB}$	ポートA, Bストローブ入力
15	ARDY	32	BRDY	ポートA, B レディ出力
16	GND	33	GND	電　源
17	〃	34	〃	

22 21 16 15 2 1 部品面
44 43 38 37 24 23 半田面

J 2

17 16 1 部品面 半田面 34 33 18

（注） V_{DD} と V_{BB} は2708タイプ EPROM用電源

J_3 （44ピン）

端子 No.	信号名（部品面）	端子 No.	信号名（半田面）	説明
1	VCC（+5V）	23	VCC（+5V）	電　源
2	〃	24	〃	
3	〃	25	〃	
4	〃	26	〃	
5		27		
6		28		
7	ARDY	29	BRDY	ポートA, B レディ出力
8	$\overline{ASTB}$	30	$\overline{BSTB}$	ポートA, B ストローブ入力
9	A_0	31	B_0	PIO2 ポートA, Bバス
10	A_1	32	B_1	
11	A_2	33	B_2	
12	A_3	34	B_3	
13	A_4	35	B_4	
14	A_5	36	B_5	
15	A_6	37	B_6	
16	A_7	38	B_7	
17		39		
18		40		
19	GND	41	GND	電　源
20	〃	42	〃	
21	〃	43	〃	
22	〃	44	〃	

付録６　端子信号説明表

<table>
<tr><td colspan="5">J_1（１００ピン）</td></tr>
<tr><td>端子
№</td><td>信号名</td><td>入力／出力</td><td>有効
レベル</td><td>説明</td></tr>
<tr><td></td><td>V_{CC}</td><td>入力</td><td></td><td>＋５Ｖ電源ライン
端子№：1,　2,　3,　51, 52, 53</td></tr>
<tr><td>５
５５</td><td>V_{DD}</td><td>〃</td><td></td><td>＋12Ｖ 電源ライン
２７０８タイプＥＰＲＯＭ用電源</td></tr>
<tr><td>１１</td><td>MRESET*</td><td>入力</td><td>Ｌ</td><td>ＣＰＵ，８２５１，ＰＩＯ，Ｆ／Ｆ を初期状態に戻します。
リセット信号入力ライン</td></tr>
<tr><td>６１</td><td>RESET *</td><td>出力</td><td>Ｌ</td><td>リセット信号</td></tr>
<tr><td>６２</td><td>ϕ</td><td>出力</td><td></td><td>システムクロック　　2.４５７６ＭHz</td></tr>
<tr><td>１６</td><td>ＤＢＩＮ*</td><td>入力</td><td>Ｌ</td><td>双方向性データバッファ制御信号
“Ｈ”：ドライブ状態　　“Ｌ”：レシーブ状態</td></tr>
<tr><td>２０</td><td>ＩＥＩ</td><td>入力</td><td>Ｈ</td><td>割り込み優先順位を形成するのに使用します。</td></tr>
<tr><td>７０</td><td>ＩＥＯ</td><td>出力</td><td>Ｈ</td><td>ＩＥＩとともに割り込み優先順位を形成するのに使用します。</td></tr>
<tr><td></td><td>A_0^*～A_{15}^*</td><td>入力／出力</td><td>Ｌ
（アドレス“１”）</td><td>１６ビットの双方向性アドレスバスで、メモリデータの交換、Ｉ／Ｏ機器データの交換に対してのアドレスを与えます。A_0^*は、最下位ビット（ＬＳＢ）です。
<table>
<tr><td>信号名</td><td>A_0^*</td><td>A_1^*</td><td>A_2^*</td><td>A_3^*</td><td>A_4^*</td><td>A_5^*</td><td>A_6^*</td><td>A_7^*</td><td>A_8^*</td><td>A_9^*</td><td>A_{10}^*</td><td>A_{11}^*</td><td>A_{12}^*</td><td>A_{13}^*</td><td>A_{14}^*</td><td>A_{15}^*</td></tr>
<tr><td>端子№</td><td>22</td><td>72</td><td>23</td><td>73</td><td>24</td><td>74</td><td>25</td><td>75</td><td>26</td><td>76</td><td>27</td><td>77</td><td>28</td><td>78</td><td>29</td><td>79</td></tr>
</table></td></tr>
<tr><td>３０</td><td>ＷＡＩＴ*</td><td>入力</td><td>Ｌ</td><td>ＣＰＵに対してアドレス付けされたメモリ、Ｉ／Ｏ機器がデータの転送準備ができていないことを示す信号です。</td></tr>
<tr><td>８０</td><td>BUSRQ *</td><td>入力</td><td>Ｌ</td><td>ＣＰＵのアドレスバス、データバス、トライステートコントロール信号線をハイインピーダンスにし、アドレスバッファ、コントロール信号用バッファをレシーブ状態にします。DMAに使用します。</td></tr>
<tr><td>８１</td><td>ＩＮＴ *</td><td>入力</td><td>Ｌ</td><td>Ｉ／Ｏ機器が割り込みを要求するときに使用します。
割り込み要求信号</td></tr>
<tr><td>３４</td><td>BUSAK *</td><td>出力</td><td>Ｌ</td><td>ＢＵＳＲＱをＣＰＵが受け付けたときに出力されます。これにより、ＣＰＵのアドレスバス、データバス、トライステートコントロール信号線がハイインピーダンスになり、外部よりこれらの信号を入力できることを示します。</td></tr>
<tr><td>８４</td><td>ＨＡＬＴ*</td><td>出力</td><td>Ｌ</td><td>ＣＰＵがＨＡＬＴ命令を実行し、プログラムの実行を停止中であることを示します。</td></tr>
<tr><td>３５</td><td>Ｍ1 *</td><td>出力</td><td>Ｌ</td><td>マシンサイクルが命令実行の中のフェッチサイクルであることを示します。また、ＩＯＲＱ* とともに生じることで、割り込みのアクノリッジサイクルを示します。</td></tr>
<tr><td>８５</td><td>ＲＦＳＨ*</td><td>出力</td><td>Ｌ</td><td>アドレスバス下位７ビット（A_6～A_0）がダイナミックメモリのためのリフレッシュアドレスを持っていることを示します。</td></tr>
<tr><td>３６</td><td>ＲＤ *</td><td>入力／出力</td><td>Ｌ</td><td>ＣＰＵが、メモリやＩ／Ｏ機器からデータを読みとろうとしていることを示します。</td></tr>
</table>

<table>
<tr><td>86</td><td>WR *</td><td>入力／出力</td><td>L</td><td>指定したメモリやI／O機器にデータを格納する際に、データバスに有効なデータを保持していることを示す信号です。</td></tr>
<tr><td>37</td><td>MREQ*</td><td>入力／出力</td><td>L</td><td>メモリリード、または、メモリライト動作に対してアドレスバスが有効なアドレスを保持していることを示します。</td></tr>
<tr><td>87</td><td>IORQ*</td><td>入力／出力</td><td>L</td><td>I／Oリード、または、I／Oライト動作に対してアドレスバスの下位8ビットが有効なアドレスを保持していることを示します。また、M1*とともに割り込みのアクノリッジを示します。</td></tr>
<tr><td></td><td>D_0*～D_7*</td><td>入力／出力</td><td>(L データ "1")</td><td>8ビットの双方向性データバスで、メモリ、I／O機器とのデータ交換に使用します。
<table><tr><td>信号名</td><td>D_0*</td><td>D_1*</td><td>D_2*</td><td>D_3*</td><td>D_4*</td><td>D_5*</td><td>D_6*</td><td>D_7*</td></tr><tr><td>端子№</td><td>42</td><td>92</td><td>43</td><td>93</td><td>44</td><td>94</td><td>45</td><td>95</td></tr></table></td></tr>
<tr><td>47
97</td><td>V_{BB}</td><td>入 力</td><td></td><td>－5V電源ライン
2708タイプEPROM用電源</td></tr>
<tr><td></td><td>GND</td><td>入 力</td><td></td><td>接地ライン
端子№：48, 49, 50, 98, 99, 100</td></tr>
</table>

J_2（34ピン）

<table>
<tr><th>端子№</th><th>信号名</th><th>入力／出力</th><th>有効レベル</th><th>説 明</th></tr>
<tr><td></td><td>V_{CC}</td><td>出 力</td><td></td><td>＋5V電源ライン
端子№：1, 2, 18, 19</td></tr>
<tr><td>21</td><td>$\overline{\text{RES}}$</td><td>入 力</td><td>L</td><td>リセット信号、キーボードスイッチの RESET キーに接続しています。</td></tr>
<tr><td>5</td><td>$\overline{PS_1}$</td><td>出 力</td><td>L</td><td>アドレス下位8ビットをデコードした信号です。
ポートアドレス：DE （システム予備）</td></tr>
<tr><td>22</td><td>$\overline{PS_2}$</td><td>出 力</td><td>L</td><td>アドレス下位8ビットをデコードした信号です。
ポートアドレス：DF （システム予備）</td></tr>
<tr><td></td><td>A_0～A_7</td><td>出 力</td><td>H</td><td>PIO_1のポートAバスラインです。
キーボードのLED用セグメント信号として出力モード（モード0）で使用
<table><tr><td>信号名</td><td>A_7</td><td>A_6</td><td>A_5</td><td>A_4</td><td>A_3</td><td>A_2</td><td>A_1</td><td>A_0</td></tr><tr><td>端子№</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td></tr></table></td></tr>
<tr><td></td><td>B_0～B_3
B_4～B_7</td><td>入 力
出 力</td><td>L</td><td>PIO_1のポートBバスラインです。
キーボードのLED用桁信号、キーボードスイッチのストローブ信号、キースイッチの入力ラインとしてビットコントロールモード（モード3）で使用
B_0～B_3 ： 入力
B_4～B_7 ： 出力
<table><tr><td>信号名</td><td>B_7</td><td>B_6</td><td>B_5</td><td>B_4</td><td>B_3</td><td>B_2</td><td>B_1</td><td>B_0</td></tr><tr><td>端子№</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr></table></td></tr>
<tr><td>14</td><td>$\overline{\text{ASTB}}$</td><td>入 力</td><td>L</td><td>周辺装置からのポートAに対するストローブ信号入力ラインです。</td></tr>
<tr><td>31</td><td>$\overline{\text{BSTB}}$</td><td>入 力</td><td>L</td><td>周辺装置からのポートBに対するストローブ信号入力ラインです。</td></tr>
</table>

15	ARDY	出　力	H	ポートAのレディ信号です。
32	BRDY	出　力	H	ポートBのレディ信号です。
	GND	出　力		接地ライン 端子No. : 16, 17, 33, 34

J_3（44ピン）

端子No.	信号名	入力／出力	有効レベル	説　明
	GND	出　力		接地ライン 端子No. : 19, 20, 21, 22, 41, 42, 43, 44
	A_0〜A_7	入力／出力	(H データ "1")	PIO$_2$のポートAバスラインです。 信号名 | A_7 | A_6 | A_5 | A_4 | A_3 | A_2 | A_1 | A_0 端子No. | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9
7	ARDY	出　力	H	ポートAのレディ信号です。
8	$\overline{ASTB}$	入　力	L	周辺装置からのポートAに対するストローブ信号入力ラインです。
	B_0〜B_7	入力／出力	(H データ "1")	PIO$_2$のポートBバスラインです。 信号名 | B_7 | B_6 | B_5 | B_4 | B_3 | B_2 | B_1 | B_0 端子No. | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31
29	BRDY	出　力	H	ポートBのレディ信号です。
30	$\overline{BSTB}$	入　力	L	周辺装置からのポートBに対するストローブ信号入力ラインです。
	Vcc	出　力		＋5Ｖ電源ライン 端子No. : 1, 2, 3, 4, 23, 24, 25, 26

端子名	信号名	入力／出力	説　明
EJ_1	REM		オーディオカセットのスタート／ストップを制御する端子です。 オーディオカセットのリモート端子と接続します。
EJ_2	AUX	出　力	オーディオカセットに録音する場合に周波数変調されたデータを出力します。カセットのAUX端子と接続します。
EJ_3	PHONE	入　力	オーディオカセットから再生する場合に可聴周波数のデータを入力します。カセットのイヤホン端子、または、外部スピーカ端子と接続します。

付録7　使用部品リスト（CPUボード）

部品番号	名称	形名・規格	備考	部品番号	名称	形名・規格	備考
U1,2	IC	MC8T26	オプション（モトローラ）	U46	IC	SN74LS14	
U3	〃	MC8T97	〃	U47	〃	SN7474	
U4	〃	MC8T28	〃	U48	〃	HA17902	
U5〜8	〃	MC8T26	〃	U49	LSI	LH−0081	Z−80 PIO
U9	〃	SN7404	オプション（T・I）	U50	〃	μPD8251C	UART
U10	〃	SN74LS08		U51	〃	LH−0081	オプション Z−80 PIO(シャープ)
U11	〃	SN74LS32		D1〜3	ダイオード	1S2075K	
U12	〃	SN74LS293		D4	LED	GL−5AR1	
U13	〃	SN74LS14		XTAL	水晶	NC18A 4.9152 M	
U14,15	〃	SN74LS42		R1〜7	抵抗	3.3 kΩ ¼w±5%	
U16	LSI	LH−7055	ROM	R8	〃	2.2 kΩ 〃	
U17	〃	〃	オプションROM(シャープ)	R9,10	〃	3.3 kΩ 〃	
U18	IC	SN7438		R11	〃	1 kΩ 〃	
U19	〃	SN74LS08		R12	〃	150Ω 〃	
U20	〃	SN74LS74		R13	〃	1 kΩ 〃	
U21	〃	SN74LS32		R14	〃	3.3 kΩ 〃	
U22	〃	SN74LS30		R15,16	〃	1 kΩ 〃	
U23	LSI	LH−2111A4	RAM	R17	〃	3.3 kΩ 〃	
U24	〃	LH−2114−3	RAM	R18,19	〃	820Ω 〃	
U25,26	〃	LH−2114	オプションRAM(シャープ)	R20	〃	330Ω 〃	
U27	IC	SN74LS74		R21 22	〃	2.2 kΩ 〃	
U28	〃	SN74LS02		R23 24	〃	3.3 kΩ 〃	
U29	〃	TC4040		R25〜28	〃	1 kΩ 〃	
U30	〃	SN74LS10		R29	〃	330Ω 〃	
U31	〃	SN74LS32		R30	〃	3.3 kΩ 〃	
U32	LSI	LH−2111A4	RAM	R31	〃	1 kΩ 〃	
U33	〃	LH−2114−3	RAM	R32	〃	10 kΩ 〃	
U34,35	〃	LH−2114	オプションRAM(シャープ)	R33〜35	〃	2.2 kΩ ¼w±2%	
U36	IC	SN74161		R36	〃	1MΩ ¼w±5%	
U37	〃	SN74LS08		R37	〃	10 kΩ ¼w±2%	
U38,39	〃	SN74LS155		R38	〃	10 kΩ ¼w±5%	
U40	〃	TC4040		R39	〃	1 kΩ ¼w±2%	
U41	〃	SN74LS32		R40〜42	〃	4.7 kΩ 〃	
U42	LSI	LH−0080	Z−80 CPU	R43	〃	100Ω ¼w±5%	
U43	IC	SN74S04		R44	〃	2.2 kΩ ¼w±2%	
U44,45	〃	SN74LS74		R45	抵抗	10 kΩ ¼w±5%	

（CPUボード）

部品番号	名　称	形名・規格	備　考
R_{46}	抵　抗	1kΩ　¼w±2%	
R_{47}	〃	220kΩ　〃	
C_1	コンデンサ	33μF　16wv	
C_2	〃	1000PF　50wv	
$C_{3,4}$	〃	330PF　〃	
C_5	〃	100PF　〃	
C_6	〃	0.01μF　〃	
C_7	〃	1μF　35wv	
C_8	〃	0.22μF　〃	
C_9	〃	1μF　〃	
C_{10}	〃	1000PF　50wv	
C_{11}	〃	0.1μF　35wv	
C_{12}	〃	0.01μF　50wv	
C_{13}	〃	1μF　35wv	
C_{14}	〃	0.022μF　50wv	
$C_{15,16}$	〃	1μF　35wv	
$C_{17,18}$	〃	33μF　25wv以上	オプション，電解コンデンサ
$C_{A\sim F}$	〃	1μF　25wv以上	オプション，タンタルコンデンサ
	〃	1μF　35wv	ノイズ防止用
ICS_1	ICソケット	C93−40−02	40ピン
$ICS_{2,3}$	〃	C93−24−02	24ピン
$ICS_{4,5}$	〃	C93−40−02	40ピン
J_1	コネクタ	4800−100−135	100ピン
J_{2-1}	〃	FAP−34−02#2	34ピンヘッダ
J_{2-2}	ソケット	FAS−34−03B	34芯フラットケーブル付
J_3	コネクタ	4600−044−112	オプション　44ピン(ケル株)
EJ_1	イヤホンジャック	X−G8514	(ケーブル付きホーンプラグ
$EJ_{2,3}$	〃	S−G8026	2.5φ，3.5φ各1本付属)
RY_1	リレー	NR−H−5V	
SW_1	スイッチ	MTM106D−R	
$P_{2,3}$	ジャンパ線		
	チェックピン	WP−2型A	

（キーボード）

部品番号	名　称	形名・規格	備　考
$U_{1\sim4}$	IC	SN75454	
U_5	〃	SN7445	
$LED_{1\sim8}$	LED	GL−9R04	
$Q_{1\sim8}$	トランジスタ	2SB561	
R_1	抵　抗	220Ω　¼w±5%	
R_2	〃	47Ω　〃	
R_3	〃	1kΩ　〃	
R_4	〃	220Ω　〃	
R_5	〃	47Ω　〃	
R_6	〃	1kΩ　〃	
R_7	〃	220Ω　〃	
R_8	〃	47Ω　〃	
R_9	〃	1kΩ　〃	
R_{10}	〃	220Ω　〃	
R_{11}	〃	47Ω　〃	
R_{12}	〃	1kΩ　〃	
R_{13}	〃	220Ω　〃	
R_{14}	〃	47Ω　〃	
R_{15}	〃	1kΩ　〃	
R_{16}	〃	220Ω　〃	
R_{17}	〃	47Ω　〃	
R_{18}	〃	1kΩ　〃	
R_{19}	〃	220Ω　〃	
R_{20}	〃	47Ω　〃	
R_{21}	〃	1kΩ　〃	
R_{22}	〃	220Ω　〃	
R_{23}	〃	47Ω　〃	
R_{24}	〃	1kΩ　〃	
C_1	コンデンサ	33μF　25wv	
C_2	〃	33μF　16wv	
J_2	コネクタ	FAP−34−02#2	34ピンヘッダ
	キースイッチ	KBD−801H	25キー
	キーパネル	A,B1対	
	ゴム足	BU692−F	

付録8　モニタプログラム リスト

```
K1.SR    ASMBL'D BY Z-80 'SSEMBLER REV-A.1     03,27,'78
         TITLE   SM-B-80T —LH-8H02

001      SM-B-80T

    1                       ;
    2                       ;         ******** SM-B-80T  V1.1 ********
    3                       ;
    4                       ;
    5                                 TITLE   SM-B-80T
    6         E000    ROM     EQU     0E000H        ;ROM STARTING ADDRESS
    7         FF00    RAM     EQU     0FF00H        ;OS RAM STARTING ADDRESS
    8         FFCC    MODE    EQU     RAM+0CCH      ;USER PROGRAM MODE
    9         FFCD    SEGBUF  EQU     MODE+1        ;SEGUMENT BUFFER
   10         FFD5    DISBUF  EQU     SEGBUF+8      ;DISPLAY BUFFER
   11         FFDD    FLAG    EQU     DISBUF+8      ;CHATTERING FLAG
   12         FFDE    REMSW   EQU     FLAG+1        ;REMOTE SWITCH FLAG
   13         FFDF    DATA    EQU     REMSW+1       ;DATA REG.
   14         FFE1    ADDR    EQU     DATA+2        ;ADDRESS REG.
   15         FFE3    SAVE    EQU     ADDR+2        ;USER REG. SAVE AREA
   16         FFFD    BCOUNT  EQU     SAVE+26       ;BREAK COUNTER
   17         FFFE    BADDR   EQU     BCOUNT+1      ;BREAK ADDRESS
   18         FFCC    STACK   EQU     MODE          ;MONITOR STACK
   19         FF9A    USER    EQU     STACK-50      ;USER STACK
   20         00D9    INT     EQU     0D9H          ;SYSTEM NMI PORT
   21         00D1    PIOAC   EQU     0D1H          ;PIO A CONTROL
   22         00D0    PIOAD   EQU     0D0H          ;PIO A DATA
   23         00D3    PIOBC   EQU     0D3H          ;PIO B CONTROL
   24         00D2    PIOBD   EQU     0D2H          ;PIO B DATA
   25         00DB    TAPEC   EQU     0DBH          ;8251 CONTROL
   26         00DA    TAPED   EQU     0DAH          ;8251 DATA
   27         00D8    SYS     EQU     0D8H          ;SYSTEM CONTROL
   28         00DC    REM     EQU     0DCH          ;REMOTE SWITCH
   29                         ORG     ROM
   30                       ;
   31                       ;
   32                       ;         ******** MAIN ********
   33                       ;
   34  E000 C303E0    MAIN:   JP      $+3
   35  E003 D3D8              OUT     (SYS),A       ;RESET E000
   36  E005 AF                XOR     A
   37  E006 0616              LD      B,22
   38  E008 21CDFF            LD      HL,SEGBUF
   39  E00B 77        MA10:   LD      (HL),A        ;CLEAR REMSW,DATA,ADDR
   40  E00C 23                INC     HL
   41  E00D 10FC              DJNZ    MA10
   42  E00F 2EFD              LD      L,BCOUNT & 0FFH
   43  E011 77                LD      (HL),A        ;CLEAR BCOUNT
   44  E012 2EF7              LD      L,SAVE+20 & 0FFH
   45  E014 77                LD      (HL),A        ;CLEAR USER IFF
   46  E015 2E9A              LD      L,USER & 0FFH ;SET UP USER STACK
   47  E017 22F3FF            LD      (SAVE),HL
   48  E01A 3ECF              LD      A,0CFH
   49  E01C D3D3              OUT     (PIOBC),A     ;SET PIO B CONTROL MODE
   50  E01E D3DB              OUT     (TAPEC),A
```

```
 51 E020 3E0F                LD      A,0FH           ;DUMMY
 52 E022 D3D3                OUT     (PIOBC),A       ;SET PIO A OUTPUT MODE
 53 E024 D3D1                OUT     (PIOAC),A
 54 E026 31CCFF     MA20:    LD      SP,STACK        ;SET UP MONITOR STACK
 55 E029 CDF7E2              CALL    DISP1           ;DISPLAY ADDR,DATA
 56 E02C CD3FE3     MA30:    CALL    KEYIN           ;KEYIN
 57 E02F 2126E0     MA50:    LD      HL,MA20         ;PUSH RETURN ADDRESS
 58 E032 E5                  PUSH    HL
 59 E033 FE10                CP      10H
 60 E035 DA1BE3              JP      C,SHIFT         ;DATA KEY
 61 E038 21E7E3              LD      HL,JPTAB        ;COMMAND KEY
 62 E03B D610                SUB     10H
 63 E03D 07                  RLCA
 64 E03E 85                  ADD     A,L
 65 E03F 6F                  LD      L,A
 66 E040 7E                  LD      A,(HL)
 67 E041 23                  INC     HL
 68 E042 66                  LD      H,(HL)
 69 E043 6F                  LD      L,A
 70 E044 E9                  JP      (HL)            ;GO TO EACH COMMAND ROUTINE
 71                 ;
 72                 ;
 73                 ;        ******** FUNCTION ********
 74                 ;
 75 E045 CD3FE3     FUNC:    CALL    KEYIN           ;NEXT KEY INPUT
 76 E048 FE11                CP      11H
 77 E04A CAC6E1              JP      Z,LOAD          ;LOAD KEY
 78 E04D FE12                CP      12H
 79 E04F CA36E2              JP      Z,STORE         ;STORE KEY
 80 E052 FE10                CP      10H
 81 E054 2859                JR      Z,RG100         ;REG' KEY
 82 E056 FE16                CP      16H
 83 E058 C0                  RET     NZ
 84 E059 21DEFF     REMOTE:  LD      HL,REMSW        ;REMOTE KEY
 85 E05C 34                  INC     (HL)            ;CHECK REMSW
 86 E05D DBDC                IN      A,(REM)         ;REMOTE ON
 87 E05F C8                  RET     Z
 88 E060 D3DC                OUT     (REM),A         ;REMOTE OFF
 89 E062 74                  LD      (HL),H          ;TURN OFF FLAG
 90 E063 C9                  RET
 91                 ;
 92                 ;
 93                 ;        ******** INTERRUPT ********
 94                 ;
 95                          ORG     ROM+66H
 96 E066 C369E0     INTER:   JP      $+3             ;NMI ENTRY ADDRESS
 97 E069 D3D8                OUT     (SYS),A         ;RESET E000
 98 E06B ED73E3FF            LD      (SAVE),SP       ;SAVE USER SP
 99 E06F 31FDFF              LD      SP,SAVE+26
100 E072 F5                  PUSH    AF
```

```
101 E073 F5               PUSH    AF              ;SAVE AF
102 E074 ED57             LD      A,I             ;SAVE I,IFF
103 E076 F5               PUSH    AF
104 E077 C5               PUSH    BC              ;SAVE BC
105 E078 D5               PUSH    DE              ;SAVE DE
106 E079 E5               PUSH    HL              ;SAVE HL
107 E07A D9               EXX
108 E07B 08               EX      AF,AF'          ;SAVE AF'
109 E07C F5               PUSH    AF
110 E07D C5               PUSH    BC              ;SAVE BC'
111 E07E D5               PUSH    DE              ;SAVE DE'
112 E07F E5               PUSH    HL              ;SAVE HL'
113 E080 3AF7FF           LD      A,(SAVE+20)
114 E083 E604             AND     4
115 E085 0F               RRCA
116 E086 0F               RRCA
117 E087 32F7FF           LD      (SAVE+20),A
118 E08A D9               EXX
119 E08B 08               EX      AF,AF'
120 E08C DDE5             PUSH    IX              ;SAVE IX
121 E08E FDE5             PUSH    IY              ;SAVE IY
122 E090 ED7BE3FF         LD      SP,(SAVE)
123 E094 E1               POP     HL              ;GET OLD PC
124 E095 ED73E3FF         LD      (SAVE),SP
125 E099 22FBFF           LD      (SAVE+24),HL    ;SAVE PC
126 E09C 22E1FF           LD      (ADDR),HL       ;DISPLAY PC
127 E09F 2AF9FF           LD      HL,(SAVE+22)
128 E0A2 22DFFF           LD      (DATA),HL       ;DISPLAY AF
129 E0A5 3ACCFF           LD      A,(MODE)        ;CHECK USER MODE
130 E0A8 3C               INC     A
131 E0A9 C25AE1           JP      NZ,EX00         ;RUN MODE
132 E0AC C326E0           JP      MA20            ;STEP MODE
133                  ;
134                  ;
135                  ;        ******** REGISTER ********
136                  ;
137 E0AF CD3FE3  RG100:   CALL    KEYIN           ;INPUT REG' KEY
138 E0B2 FE08             CP      8
139 E0B4 D8               RET     C
140 E0B5 FE10             CP      10H
141 E0B7 D0               RET     NC
142 E0B8 C608             ADD     A,8             ;ADD OFFSET
143 E0BA 1806             JR      RG60
144 E0BC CD3FE3  REG:     CALL    KEYIN           ;INPUT REG. KEY
145 E0BF FE10             CP      10H
146 E0C1 D0               RET     NC              ;IF COMMAND KEY, IGNORE
147 E0C2 47      RG60:    LD      B,A
148 E0C3 87               ADD     A,A
149 E0C4 119DE3           LD      DE,SYMTAB
150 E0C7 83               ADD     A,E
```

```
151 E0C8 5F              LD      E,A
152 E0C9 21DCFF          LD      HL,DISBUF+7
153 E0CC 1A              LD      A,(DE)
154 E0CD E6F0            AND     0F0H
155 E0CF 0F              RRCA
156 E0D0 0F              RRCA
157 E0D1 0F              RRCA
158 E0D2 0F              RRCA
159 E0D3 C60A            ADD     A,0AH        ;GET 1'ST CHARACTER
160 E0D5 77              LD      (HL),A       ;STORE IT TO DISPLAY BUFFER
161 E0D6 2B              DEC     HL
162 E0D7 1A              LD      A,(DE)
163 E0D8 E60F            AND     0FH
164 E0DA C60A            ADD     A,0AH        ;GET 2'ND CHARACTER
165 E0DC 77              LD      (HL),A       ;STORE IT
166 E0DD 2B              DEC     HL
167 E0DE 3615            LD      (HL),15H     ;STORE SPACE
168 E0E0 2B              DEC     HL
169 E0E1 3617            LD      (HL),17H     ;STORE '-'
170 E0E3 13              INC     DE
171 E0E4 1A              LD      A,(DE)
172 E0E5 21E3FF          LD      HL,SAVE
173 E0E8 85              ADD     A,L
174 E0E9 6F              LD      L,A          ;GET ADDR. OF REG. SAVE AREA
175 E0EA 54              LD      D,H
176 E0EB 5D              LD      E,L
177 E0EC 7E              LD      A,(HL)
178 E0ED 23              INC     HL
179 E0EE 66              LD      H,(HL)
180 E0EF 6F              LD      L,A
181 E0F0 78              LD      A,B
182 E0F1 FE05            CP      5            ;1 BYTE REG.
183 E0F3 3802            JR      C,RG10
184 E0F5 2600            LD      H,0
185 E0F7 22DFFF   RG10:  LD      (DATA),HL    ;STORE CONTAINT OF REG. TO DATA REG.
186 E0FA CDFCE2          CALL    DISP2
187 E0FD CD3FE3   RG20:  CALL    KEYIN        ;INPUT NEXT KEY
188 E100 FE13            CP      13H          ;IF RUN KEY, RETURN
189 E102 C8              RET     Z
190 E103 FE15            CP      15H
191 E105 2814            JR      Z,RG40       ;WRITE KEY
192 E107 FE11            CP      11H
193 E109 281C            JR      Z,RG70       ;INCR KEY
194 E10B FE12            CP      12H
195 E10D 2821            JR      Z,RG80       ;DECR KEY
196 E10F FE10            CP      10H
197 E111 3026            JR      NC,RG90
198 E113 CD1BE3          CALL    SHIFT        ;DATA KEY
199 E116 CDFCE2          CALL    DISP2        ;DISPLAY DATA REG.
200 E119 18F2            JR      RG20
```

```
201 E11B 2ADFFF      RG40:   LD      HL,(DATA)        ;LOAD DATA TO HL
202 E11E EB                  EX      DE,HL
203 E11F 73                  LD      (HL),E           ;STORE LOW BYTE TO ITS SAVE AREA
204 E120 78                  LD      A,B
205 E121 FE05                CP      5
206 E123 3002                JR      NC,RG70          ;IF 1 BYTE REG, SKIP
207 E125 23                  INC     HL
208 E126 72                  LD      (HL),D           ;STORE HIGH BYTE
209 E127 78          RG70:   LD      A,B
210 E128 3C                  INC     A                ;INREMENTE REG. NO.
211 E129 FE18                CP      18H
212 E12B 3895                JR      C,RG60
213 E12D AF                  XOR     A                ;IF OVER RANGE, CLEAR REG. NO.
214 E12E 1892                JR      RG60
215 E130 78          RG80:   LD      A,B
216 E131 3D                  DEC     A
217 E132 F2C2E0              JP      P,RG60
218 E135 3E17                LD      A,23
219 E137 1889                JR      RG60
220 E139 E1          RG90:   POP     HL
221 E13A F5                  PUSH    AF
222 E13B CD24E3              CALL    SEGCON
223 E13E F1                  POP     AF               ;RECOVER COMMAND KEY
224 E13F C32FE0              JP      MA50
225                  ;
226                  ;
227                  ;       ******** RUN,STEP ********
228                  ;
229 E142 3EFF        STEP:   LD      A,0FFH           ;STEP COMMAND ROUTINE
230 E144 32CCFF      EX50:   LD      (MODE),A         ;SET USER PROGRAM MODE
231 E147 1824                JR      EX40
232 E149 2AE1FF      RUN:    LD      HL,(ADDR)        ;RUN COMMAND ROUTINE
233 E14C 22FBFF              LD      (SAVE+24),HL     ;SET UP PC
234 E14F 3AFDFF              LD      A,(BCOUNT)       ;CHECK BCOUNT
235 E152 A7                  AND     A
236 E153 28EF                JR      Z,EX50           ;IF BCOUNT IS NON ZERO, TRACE MODE
237 E155 3E01                LD      A,1              ;BP IS ACTIVE
238 E157 32CCFF      EXEC:   LD      (MODE),A         ;SET USER PROGRAM MODE
239 E15A 2AFBFF      EX00:   LD      HL,(SAVE+24)     ;GET CURRENT USER'S PC
240 E15D ED5BFEFF            LD      DE,(BADDR)       ;GET BREAK ADDRESS
241 E161 B7                  OR      A
242 E162 ED52                SBC     HL,DE            ;CHECK THEM
243 E164 2007                JR      NZ,EX40          ;IF NOT EQUALE, COUNTINE
244 E166 21FDFF              LD      HL,BCOUNT        ;IF EQUALE, DECREMENT BCOUNT
245 E169 35                  DEC     (HL)
246 E16A CA26E0              JP      Z,MA20           ;IF ZERO, BREAK
247 E16D 2AFBFF      EX40:   LD      HL,(SAVE+24)     ;GET PC
248 E170 E5                  PUSH    HL
249 E171 7E                  LD      A,(HL)
250 E172 FECB                CP      0CBH
```

```
251 E174 2808            JR    Z,EX41
252 E176 E60F            AND   0FH
253 E178 FE0D            CP    0DH
254 E17A 200A            JR    NZ,EX42
255 E17C 7E              LD    A,(HL)
256 E17D FEDD            CP    0DDH
257 E17F 3805            JR    C,EX42
258 E181 21F7FF   EX41:  LD    HL,SAVE+20
259 E184 CBF6            SET   6,(HL)
260 E186 E1       EX42:  POP   HL
261 E187 ED7BE3FF        LD    SP,(SAVE)     ;GET SP
262 E18B E5              PUSH  HL            ;PUSH USER'S PC ON USER'S STACK
263 E18C 2AF9FF          LD    HL,(SAVE+22)  ;GET AF
264 E18F E5              PUSH  HL            ;PUSH AF TOO
265 E190 ED73E3FF        LD    (SAVE),SP     ;SAVE SP
266 E194 31E5FF          LD    SP,SAVE+2     ;LOAD SAVE ADDRESS TO SP
267 E197 FDE1            POP   IY            ;RESTORE IY
268 E199 DDE1            POP   IX            ;RESTORE IX
269 E19B D9              EXX
270 E19C 08              EX    AF,AF'
271 E19D E1              POP   HL            ;RESTORE HL'
272 E19E D1              POP   DE            ;RESTORE DE'
273 E19F C1              POP   BC            ;RESTORE BC'
274 E1A0 F1              POP   AF            ;RESTORE AF'
275 E1A1 D9              EXX
276 E1A2 08              EX    AF,AF'
277 E1A3 E1              POP   HL            ;RESTORE HL
278 E1A4 D1              POP   DE            ;RESTORE DE
279 E1A5 C1              POP   BC            ;RESTORE BC
280 E1A6 F1              POP   AF
281 E1A7 ED47            LD    I,A           ;RESTORE I
282 E1A9 ED7BE3FF        LD    SP,(SAVE)     ;RESTORE SP
283 E1AD 3ACCFF          LD    A,(MODE)          ;GET MODE
284 E1B0 3001            JR    NC,EX10       ;CHECK USER'S IFF
285 E1B2 FB              EI
286 E1B3 2009     EX10:  JR    NZ,EX30
287 E1B5 A7              AND   A             ;CHECK USER MODE
288 E1B6 2804            JR    Z,EX20
289 E1B8 F1              POP   AF
290 E1B9 D3D9            OUT   (INT),A
291 E1BB C9              RET                 ;RETURN TO USER PROGRAM
292 E1BC F1       EX20:  POP   AF
293 E1BD C9              RET                 ;RETURN TO USER PROGRAM
294 E1BE A7       EX30:  AND   A
295 E1BF 28FB            JR    Z,EX20
296 E1C1 F1              POP   AF
297 E1C2 D3D9            OUT   (INT),A
298 E1C4 00              NOP
299 E1C5 C9              RET                 ;RETURN TO PROGRAM
300
```

```
301                       ;
302                       ;          ******** LOAD ********
303                       ;
304 E1C6 3E40     LOAD:   LD         A,40H          ;LOAD COMMAND ROUTINE
305 E1C8 D3DB             OUT        (TAPEC),A      ;RESET 8251
306 E1CA 3ECD             LD         A,0CDH         ;SET UP 8251
307 E1CC D3DB             OUT        (TAPEC),A
308 E1CE D3DC             OUT        (REM),A        ;REMOTE ON
309 E1D0 CDB5E2           CALL       WAIT5          ;WAIT 5 SEC.
310 E1D3 3E04             LD         A,4
311 E1D5 D3DB             OUT        (TAPEC),A      ;RECEIVE ENABLE
312 E1D7 1E01             LD         E,1            ;INITIALIZE BLOCK NO.
313 E1D9 0E00     LD10:   LD         C,0
314 E1DB CD29E2           CALL       TPIN           ;READ BLOCK NO.
315 E1DE BB               CP         E              ;CHECK IT
316 E1DF 2033             JR         NZ,ERROR                 ;IF NOT EQUALE, ERROR
317 E1E1 CD29E2           CALL       TPIN
318 E1E4 47               LD         B,A            ;READ BLOCK LENGTH
319 E1E5 CD29E2           CALL       TPIN
320 E1E8 57               LD         D,A            ;READ BLOCK TYPE
321 E1E9 CD29E2           CALL       TPIN
322 E1EC 67               LD         H,A            ;READ LOAD ADDRESS (HI)
323 E1ED CD29E2           CALL       TPIN
324 E1F0 6F               LD         L,A            ;READ LOAD ADDRESS (LOW)
325 E1F1 78               LD         A,B
326 E1F2 A7               AND        A
327 E1F3 2811             JR         Z,LD40         ;IF BLOCK LENGTH = 0, SKIP
328 E1F5 7B               LD         A,E
329 E1F6 3D               DEC        A
330 E1F7 2003             JR         NZ,LD20
331 E1F9 22E1FF           LD         (ADDR),HL      ;IF 1'ST BLOCK, SET LOAD ADDRESS
332 E1FC CD29E2   LD20:   CALL       TPIN           ;READ DATA
333 E1FF 77               LD         (HL),A         ;AND STORE IT TO MEMORY
334 E200 22DFFF           LD         (DATA),HL      ;SET CURRENT LOAD ADDRESS
335 E203 23               INC        HL
336 E204 10F6             DJNZ       LD20
337 E206 CD29E2   LD40:   CALL       TPIN           ;READ CHECK SUM
338 E209 79               LD         A,C
339 E20A A7               AND        A
340 E20B 2007             JR         NZ,ERROR       ;IF C IS NON ZERO, CHECK SUM ERROR
341 E20D 1C               INC        E
342 E20E 15               DEC        D              ;IF BLOCK TYPE = 1, DATA BLOCK
343 E20F 28C8             JR         Z,LD10
344 E211 14               INC        D              ;IF BLOCK TYPE = 0, END OF BLOCK
345 E212 280E             JR         Z,LD30
346 E214 0608     ERROR:  LD         B,8            ;ERROR MESSAGE ROUTINE
347 E216 21CDFF           LD         HL,SEGBUF
348 E219 3640     ER10:   LD         (HL),40H       ;DISPLAY '----' '----'
349 E21B 23               INC        HL
350 E21C 10FB             DJNZ       ER10
```

```
351 E21E 212CF0              LD      HL,MA30
352 E221 E3                  EX      (SP),HL
353 E222 AF         LD30:    XOR     A
354 E223 32DEFF              LD      (REMSW),A
355 E226 DBDC                IN      A,(REM)         ;REMOTE OFF
356 E228 C9                  RET
357 E229 DBDB       TPIN:    IN      A,(TAPEC)       ;RECEIVE READY?
358 E22B CB4F                BIT     1,A
359 E22D 28FA                JR      Z,TPIN          ;NO, CHECK AGAIN
360 E22F DBDA                IN      A,(TAPED)       ;INPUT DATA
361 E231 F5                  PUSH    AF
362 E232 81                  ADD     A,C
363 E233 4F                  LD      C,A             ;RENEW CHECK SUM
364 E234 F1                  POP     AF
365 E235 C9                  RET
366                 ;
367                 ;
368                 ;        ******** STORE ********
369                 ;
370 E236 3E40       STORE:   LD      A,40H           ;CASSETTE TAPE STORE ROUTINE
371 E238 D3DB                OUT     (TAPEC),A       ;RESET 8251
372 E23A 3ECE                LD      A,0CEH
373 E23C D3DB                OUT     (TAPEC),A       ;SET UP 8251
374 E23E 3E01                LD      A,1
375 E240 D3DB                OUT     (TAPEC),A       ;TRANSMIT ENABLE
376 E242 2ADFFF              LD      HL,(DATA)
377 E245 ED5BE1FF            LD      DE,(ADDR)
378 E249 AF                  XOR     A
379 E24A ED52                SBC     HL,DE
380 E24C D8                  RET     C
381 E24D D3DC                OUT     (REM),A         ;REMOTE ON
382 E24F 23                  INC     HL              ;STORE BYTE COUNT
383 E250 E5                  PUSH    HL
384 E251 063C                LD      B,60
385 E253 CDB7E2              CALL    WAIT            ;WAIT 30 SEC.
386 E256 1E01                LD      E,1
387 E258 53         ST50:    LD      D,E
388 E259 CDC2E2              CALL    TPOUT           ;WRITE BLOCK NO.
389 E25C E1                  POP     HL
390 E25D 01FF00              LD      BC,255          ;MAX. LENGTH IN A BLOCK
391 E260 AF                  XOR     A
392 E261 ED42                SBC     HL,BC
393 E263 3008                JR      NC,ST10
394 E265 09                  ADD     HL,BC
395 E266 85                  ADD     A,L
396 E267 47                  LD      B,A
397 E268 2E00                LD      L,0
398 E26A 282E                JR      Z,ST20
399 E26C 3E                  DEFB    3EH             ;SKIP NEXT INSTRUCTION
400 E26D 41          ST10:    LD      B,C
```

```
401 E26E F5        ST30:   PUSH    HL
402 E26F 4B                LD      C,E
403 E270 50                LD      D,B
404 E271 CDC2E2            CALL    TPOUT          ;WRITE BLOCK LENGTH
405 E274 1601              LD      D,1
406 E276 CDC2E2            CALL    TPOUT          ;WRITE BLOCK TYPE
407 E279 2AE1FF            LD      HL,(ADDR)
408 E27C 54                LD      D,H
409 E27D CDC2E2            CALL    TPOUT          ;WRITE LOAD ADDRESS (HI)
410 E280 55                LD      D,L
411 E281 CDC2E2            CALL    TPOUT          ;WRITE LOAD ADDRESS (LOW)
412 E284 56        ST40:   LD      D,(HL)
413 E285 CDC2E2            CALL    TPOUT          ;WRITE DATA
414 E288 23                INC     HL
415 E289 10F9              DJNZ    ST40
416 E28B 78                LD      A,B
417 E28C 91                SUB     C
418 E28D 57                LD      D,A
419 E28E CDC2E2            CALL    TPOUT          ;WRITE CHECK SUM
420 E291 22E1FF            LD      (ADDR),HL
421 E294 CDB5E2            CALL    WAIT5          ;WAIT 5 SEC.
422 E297 1C                INC     E
423 E298 18BE              JR      ST50
424 E29A 55        ST20:   LD      D,L
425 E29B 0604              LD      B,4
426 E29D CDC2E2    ST60:   CALL    TPOUT          ;WRITE END BLOCK
427 E2A0 10FB              DJNZ    ST60
428 E2A2 78                LD      A,B
429 E2A3 93                SUB     E
430 E2A4 57                LD      D,A
431 E2A5 CDC2E2            CALL    TPOUT          ;WRITE CHECK SUM
432 E2A8 CDB5E2            CALL    WAIT5          ;WAIT 5 SEC.
433 E2AB 3E08              LD      A,8
434 E2AD D3DB              OUT     (TAPEC),A      ;WRITE SPACE
435 E2AF CDB5E2            CALL    WAIT5          ;WAIT 5 SEC.
436 E2B2 C322E2            JP      LD30
437 E2B5 060A      WAIT5:  LD      B,10           ;5 SEC. WAIT ROUTINE
438 E2B7 219DB8    WAIT:   LD      HL,47261       ;0.5 SEC. WAIT ROUTINE
439 E2BA 2B        WA10:   DEC     HL
440 E2BB 7C                LD      A,H
441 E2BC B5                OR      L
442 E2BD 20FB              JR      NZ,WA10
443 E2BF 10F6              DJNZ    WAIT
444 E2C1 C9                RET
445 E2C2 DBDB      TPOUT:  IN      A,(TAPEC)      ;TRANSMIT READY?
446 E2C4 0F                RRCA
447 E2C5 30FB              JR      NC,TPOUT       ;NO, WAIT
448 E2C7 7A                LD      A,D
449 E2C8 D3DA              OUT     (TAPED),A      ;OUTPUT DATA
450 E2CA 81                ADD     A,C
```

```
451 E2CB 4F               LD      C,A               ;RENEW CHECK SUM
452 E2CC C9               RET
453                ;
454                ;
455                ;          ******** ADRSET ********
456                ;
457 E2CD 2ADFFF    ADRSET: LD      HL,(DATA)        ;ADDRESS SET ROUTINE
458 E2D0 22E1FF            LD      (ADDR),HL
459 E2D3 181C              JR      IR10
460                ;
461                ;
462                ;          ******** DECR ********
463                ;
464 E2D5 2AE1FF    DECR:   LD      HL,(ADDR)        ;DECREMENT & READ MEMOTY ROUTINE
465 E2D8 2B                DEC     HL
466 E2D9 1816              JR      IR10
467 E2DB 3ADFFF    READ:   LD      A,(DATA)
468 E2DE 32E0FF            LD      (DATA+1),A       ;SHIFT 1 BYTE DATA REG.
469 E2E1 7E                LD      A,(HL)           ;LOAD MEMORY DATA
470 E2E2 32DFFF            LD      (DATA),A         ;STORE IT TO DATA REG.
471 E2E5 C9                RET
472                ;
473                ;
474                ;          ******** WRITE ********
475                ;
476 E2E6 2AE1FF    WRITE:  LD      HL,(ADDR)        ;WRITE ROUTINE
477 E2E9 3ADFFF            LD      A,(DATA)
478 E2EC 77                LD      (HL),A
479                ;
480                ;
481                ;          ******** INCR ********
482                ;
483 E2ED 2AE1FF    INCR:   LD      HL,(ADDR)        ;INCREMENT & READ MEMORY ROUTINE
484 E2F0 23                INC     HL
485 E2F1 CDDBE2    IR10:   CALL    READ             ;MEMORY READ
486 E2F4 22E1FF            LD      (ADDR),HL
487                ;
488                ;
489                ;          ******** DISP1 ********
490                ;
491 E2F7 D9        DISP1:  EXX                      ;DISPLAY DATA REG. ROUTINE
492 E2F8 0604              LD      B,4
493 E2FA 1803              JR      DP10
494                ;
495                ;
496                ;          ******** DISP2 ********
497                ;
498 E2FC D9        DISP2:  EXX                      ;DISPLAY DATA & ADDR
499 E2FD 0602              LD      B,2
500 E2FF 21DFFF    DP10:   LD      HL,DATA          ;HL IS POINTER TO DATA REG.
```

```
501 E302 11D5FF             LD      DE,DISBUF       ;DE IS POINTER TO DISBUF
502 E305 7E        DP20:    LD      A,(HL)          ;GET DATA REG.
503 E306 E60F               AND     0FH             ;MASK OFF UPPER 4 BITS
504 E308 12                 LD      (DE),A          ;STORE IT TO DISPLAY BUFFER
505 E309 7E                 LD      A,(HL)          ;RECOVER IT
506 E30A E6F0               AND     0F0H            ;MASK OFF LOWER 4 BITS
507 E30C 0F                 RRCA                    ;ROTATE RIGHT 4 BITS
508 E30D 0F                 RRCA
509 E30E 0F                 RRCA
510 E30F 0F                 RRCA
511 E310 13                 INC     DE
512 E311 12                 LD      (DE),A          ;STORE IT
513 E312 13                 INC     DE
514 E313 23                 INC     HL
515 E314 10EF               DJNZ    DP20
516 E316 CD24E3             CALL    SEGCON          ;SEGUMENT CONVERT
517 E319 D9                 EXX
518 E31A C9                 RET
519                ;
520                ;
521                ;        ******** SHIFT ********
522                ;
523 E31B 21DFFF    SHIFT:   LD      HL,DATA         ;SHIFT DIGIT RIGHT DATA REG.
524 E31E ED6F               RLD
525 E320 23                 INC     HL
526 E321 ED6F               RLD
527 E323 C9                 RET
528                ;
529                ;
530                ;        ******** SEGCON ********
531                ;
532 E324 0608      SEGCON:  LD      B,8             ;SET UP COUNTER
533 E326 21D5FF             LD      HL,DISBUF
534 E329 11CDFF             LD      DE,SEGBUF
535 E32C 7E        SC10:    LD      A,(HL)
536 E32D E5                 PUSH    HL
537 E32E 21CDE3             LD      HL,SEGTAB       ;HL IS POINTER TO SEGUMENT TABLE
538 E331 85                 ADD     A,L
539 E332 6F                 LD      L,A
540 E333 4E                 LD      C,(HL)
541 E334 1A                 LD      A,(DE)
542 E335 E680               AND     80H             ;NON EFFECTIVE DECIMAL POINT
543 E337 B1                 OR      C
544 E338 12                 LD      (DE),A
545 E339 E1                 POP     HL
546 E33A 23                 INC     HL
547 E33B 13                 INC     DE
548 E33C 10EE               DJNZ    SC10
549 E33E C9                 RET
550
```

```
551                     ;
552                     ;       ******** KEYIN ********
553                     ;
554 E33F D9      KEYIN: EXX                      ;KEY INPUT & LED DISPLAY ROUTINE
555 E340 CD70E3  KI60:  CALL    SCAN             ;KEY INPUT?
556 E343 2826           JR      Z,KI10           ;NO, CLEAR FLAG
557 E345 0604    KI30:  LD      B,4              ;WAIT FOR CHATTERING TIME
558 E347 CD70E3  KI20:  CALL    SCAN
559 E34A 10FB           DJNZ    KI20
560 E34C 281D           JR      Z,KI10
561 E34E 5F             LD      E,A              ;SAVE KEY NO.
562 E34F 3ADDFF         LD      A,(FLAG)
563 E352 A7             AND     A
564 E353 20F0           JR      NZ,KI30          ;IF FLAG IS NON ZERO, WAIT AGAIN
565 E355 7B             LD      A,E
566 E356 32DDFF         LD      (FLAG),A         ;TURN ON FLAG
567 E359 E60F           AND     0FH
568 E35B 0E00           LD      C,0
569 E35D 0F      KI40:  RRCA
570 E35E 3803           JR      C,KI50
571 E360 0C             INC     C
572 E361 18FA           JR      KI40
573 E363 7B      KI50:  LD      A,E
574 E364 E670           AND     70H
575 E366 0F             RRCA
576 E367 0F             RRCA
577 E368 B1             OR      C                ;A IS KEY SEQ. NO.
578 E369 D9             EXX
579 E36A C9             RET
580 E36B 32DDFF  KI10:  LD      (FLAG),A
581 E36E 18D0           JR      KI60
582 E370 1600    SCAN:  LD      D,0              ;SCAN LED & KEY BOAD
583 E372 0E70           LD      C,70H            ;SET UP DIGIT COUNTER
584 E374 21D4FF         LD      HL,SEGBUF+7      ;HL IS POINTER TO SEGUMENT BUFFER
585 E377 3E80    SN20:  LD      A,80H
586 E379 D3D2           OUT     (PIOBD),A        ;DISABLE DIGIT SIGNAL
587 E37B 7E             LD      A,(HL)           ;LOAD SEGUMENT DATA
588 E37C 2B             DEC     HL
589 E37D D3D0           OUT     (PIOAD),A        ;OUTPUT IT
590 E37F 79             LD      A,C
591 E380 D3D2           OUT     (PIOBD),A        ;OUTPUT DIGIT DATA
592 E382 DBD2           IN      A,(PIOBD)        ;INPUT KEY DATA
593 E384 2F             CPL
594 E385 E60F           AND     0FH
595 E387 2802           JR      Z,SN10
596 E389 B1             OR      C                ;IF GET ANY KEY, SAVE IT
597 E38A 57             LD      D,A
598 E38B 3E28    SN10:  LD      A,40             ;WAIT
599 E38D 3D      SN30:  DEC     A
600 E38E 20FD           JR      NZ,SN30
```

```
601 E390 79                LD      A,C
602 E391 D610              SUB     10H
603 E393 4F                LD      C,A             ;NEXT DIGIT
604 E394 30E1              JR      NC,SN20
605 E396 3E80              LD      A,80H
606 E398 D3D2              OUT     (PIOBD),A       ;DISABLE DIGIT SIGNAL
607 E39A 7A                LD      A,D
608 E39B A7                AND     A               ;RESTORE KEY DATA
609 E39C C9                RET
610                 ;
611                 ;
612                 ;      ******** SYMTAB ********
613                 ;
614 E39D 82         SYMTAB: DEFB   82H             ;PC
615 E39E 18                DEFB    24
616 E39F F8                DEFB    0F8H            ;SP
617 E3A0 00                DEFB    0
618 E3A1 E9                DEFB    0E9H            ;IX
619 E3A2 04                DEFB    4
620 E3A3 EA                DEFB    0EAH            ;IY
621 E3A4 02                DEFB    2
622 E3A5 10                DEFB    10H             ;BA
623 E3A6 1B                DEFB    27
624 E3A7 12                DEFB    12H             ;BC
625 E3A8 1A                DEFB    26
626 E3A9 ED                DEFB    0EDH            ;I
627 E3AA 15                DEFB    21
628 E3AB E5                DEFB    0E5H            ;IF
629 E3AC 14                DEFB    20
630 E3AD 6D                DEFB    6DH             ;H
631 E3AE 0F                DEFB    15
632 E3AF 7D                DEFB    7DH             ;L
633 E3B0 0E                DEFB    14
634 E3B1 0D                DEFB    0DH             ;A
635 E3B2 17                DEFB    23
636 E3B3 1D                DEFB    1DH             ;B
637 E3B4 13                DEFB    19
638 E3B5 2D                DEFB    2DH             ;C
639 E3B6 12                DEFB    18
640 E3B7 3D                DEFB    3DH             ;D
641 E3B8 11                DEFB    17
642 E3B9 4D                DEFB    4DH             ;E
643 E3BA 10                DEFB    16
644 E3BB 5D                DEFB    5DH             ;F
645 E3BC 16                DEFB    22
646 E3BD 6C                DEFB    6CH             ;H'
647 E3BE 07                DEFB    7
648 E3BF 7C                DEFB    7CH             ;L'
649 E3C0 06                DEFB    6
650 E3C1 0C                DEFB    0CH             ;A'
```

```
651 E3C2 0D                 DEFB    13
652 E3C3 1C                 DEFB    1CH             ;B'
653 E3C4 0B                 DEFB    11
654 E3C5 2C                 DEFB    2CH             ;C'
655 E3C6 0A                 DEFB    10
656 E3C7 3C                 DEFB    3CH             ;D'
657 E3C8 09                 DEFB    9
658 E3C9 4C                 DEFB    4CH             ;E'
659 E3CA 08                 DEFB    8
660 E3CB 5C                 DEFB    5CH             ;F'
661 E3CC 0C                 DEFB    12
662                 ;
663                 ;
664                 ;       ******** SEGTAB ********
665                 ;
666 E3CD 5C         SEGTAB: DEFB    5CH             ;0
667 E3CE 06                 DEFB    6               ;1
668 E3CF 5B                 DEFB    5BH             ;2
669 E3D0 4F                 DEFB    4FH             ;3
670 E3D1 66                 DEFB    66H             ;4
671 E3D2 6D                 DEFB    6DH             ;5
672 E3D3 7D                 DEFB    7DH             ;6
673 E3D4 27                 DEFB    27H             ;7
674 E3D5 7F                 DEFB    7FH             ;8
675 E3D6 6F                 DEFB    6FH             ;9
676 E3D7 77                 DEFB    77H             ;A
677 E3D8 7C                 DEFB    7CH             ;B
678 E3D9 39                 DEFB    39H             ;C
679 E3DA 5E                 DEFB    5EH             ;D
680 E3DB 79                 DEFB    79H             ;E
681 E3DC 71                 DEFB    71H             ;F
682 E3DD 74                 DEFB    74H             ;H
683 E3DE 38                 DEFB    38H             ;L
684 E3DF 73                 DEFB    73H             ;P
685 E3E0 76                 DEFB    76H             ;X
686 E3E1 6E                 DEFB    6EH             ;Y
687 E3E2 40                 DEFB    40H             ;-
688 E3E3 20                 DEFB    20H             ;'
689 E3E4 00                 DEFB    0               ;SPACE
690 E3E5 06                 DEFB    6               ;I
691 E3E6 6D                 DEFB    6DH             ;S
692                 ;
693                 ;
694                 ;       ******** JPTAB ********
695                 ;
696 E3E7 BCE0       JPTAB:  DEFW    REG             ;REGISTER DISPLAY ROUTINE
697 E3E9 EDE2               DEFW    INCR            ;INCREMENT & READ MEMOTY ROUTINE
698 E3EB D5E2               DEFW    DECR            ;DECREMENT & READ MEMORY ROUTINE
699 E3ED 49E1               DEFW    RUN             ;EXECUTE USER'S PROGRAM ROUTINE
700 E3EF 42F1               DEFW    STEP            ;ONE STEP EXECUTE USER'S PROGRAM
                                                     ROUTINE
```

```
701 E3F1 E6E2          DEFW   WRITE       ;WRITE & READ MEMORY ROUTINE
702 E3F3 CDE2          DEFW   ADRSET      ;ADDRESS SET ROUTINE
703 E3F5 45E0          DEFW   FUNC        ;SHIFT KEY ROUTINE
704 E3F7               END
```

SM-B-80T SYMBOL TABLE

```
ADDR   -FFF1    ADRSET-E2CD    BADDR -FFFE     BCOUNT-FFFD
DATA   -FFDF    DECR   -E2D5   DISBUF-FFD5     DISP1 -E2F7
DISP2  -E2FC    DP10   -E2FF   DP20  -E305     ER10  -E219
ERROR  -E214    EX00   -E15A   EX10  -E1B3     EX20  -E1BC
EX30   -E1BE    EX40   -E16D   EX41  -E181     EX42  -E186
EX50   -E144    EXEC   -E157   FLAG  -FFDD     FUNC  -E045
INCR   -E2ED    INT    -00D9   INTER -E066     IR10  -E2F1
JPTAB  -E3E7    KEYIN  -E33F   KI10  -E36B     KI20  -E347
KI30   -E345    KI40   -E35D   KI50  -E363     KI60  -E340
LD10   -E1D9    LD20   -E1FC   LD30  -E222     LD40  -E206
LOAD   -E1C6    MA10   -E00B   MA20  -E026     MA30  -E02C
MA50   -E02F    MAIN   -E000   MODE  -FFCC     PIOAC -00D1
PIOAD  -00D0    PIOBC  -00D3   PIOBD -00D2     RAM   -FF00
READ   -E2DB    REG    -E0BC   REM   -00DC     REMOTE-E059
REMSW  -FFDE    RG10   -E0F7   RG100 -E0AF     RG20  -E0FD
RG40   -E11B    RG60   -E0C2   RG70  -E127     RG80  -E130
RG90   -E139    ROM    -E000   RUN   -E149     SAVE  -FFE3
SC10   -E32C    SCAN   -E370   SEGBUF-FFCD     SFGCON-E324
SEGTAB-E3CD     SHIFT  -E31B   SN10  -E38B     SN20  -E377
SN30   -E38D    ST10   -E26D   ST20  -E29A     ST30  -E26E
ST40   -E284    ST50   -E258   ST60  -E29D     STACK -FFCC
STEP   -E142    STORE  -E236   SYMTAB-E39D     SYS   -00D8
TAPEC  -00DB    TAPED  -00DA   TPIN  -E229     TPOUT -E2C2
USER   -FF9A    WA10   -E2BA   WAIT  -E2B7     WAIT5 -E2B5
WRITE  -E2E6
```

Z-80
クロスアセンブラマニュアル

8

目　　　　次

Z-80　クロス・アセンブラ・マニュアル

1. 概　　要

Z－80クロス・アセンブラ(Z80A)は、ミニ・コンピュータ'NOVA'のDOS(ディスク・オペレーティング・システム)下で実行できる、Z80ソース・プログラムのアセンブル用プログラムである。

クロス・アセンブラは、NOVAのアセンブリ語で書かれていて、全ステップ数はデータ・エリアも含め約7KWである。　NOVAのDOSサイズにも依るが、主記憶部の必要容量として、16～24KW程度あれば実行可能である。

Z－80クロス・アセンブラはNOVAの'FDOS'　'RDOS'　'MRDOS'のいずれの管理下でも、またNOVA01、02、3のいずれの機種においても実行できるようになっている。

アセンブラの機能は、次の如くである。

1. オブジェクト・ファイルの作成と出力
2. アセンブル・リスト及びシンボル・テーブルの作成と出力
3. クロス・リファレンス・リストの作成と出力
4. 各リストのディスク・ファイルの作成

入力・出力装置は、NOVAのDOSが管理するすべての装置を利用できる。

1. コンソール　(TTY、CRT)
2. 紙テープ・リーダ　パンチャ　(PTR　、PTP)
3. 紙カード・リーダ　(CDR　)
4. ライン・プリンタ　(LPT)

アセンブルすべきソース・ファイルとして、次のいずれかを用意する必要がある。

1. オフラインで作成された紙テープ・ソース
2. テキスト・エディタなどで作成したソース・ファイル
3. リンクされたディスク・ファイル
4. IBMカード・ソースなど

アセンブル後作成されるオブジェクトは、直接紙テープに出力する以外に、必要なら一旦ディスク・ファイルとして作成し、保存しておくことも可能である。このディスク・ファイルは、適時他の形体(紙テープ、磁気テープ)で出力できる。

他のファイル(リストなど)も、同様にできる。

2. アセンブラ（Ｚ８０Ａ）の機能

クロス・アセンブラ（Ｚ８０Ａ）は、次のような機能を有している。

○　Ｚ－８０のアセンブラ語（ザイログ社オリジナル）で書かれたソース・プログラムから、所定の形式（インテル・フォーマット）のオブジェクト・ファイルを作成し、出力する。

○　ソース・プログラムに、エラー・コード、ステートメント番号、アドレス、マシン・コードなどを付し、リポートするアセンブル・リストを出力する。

○　ラベル（シンボル）を一覧表にしたシンボル・テーブルを作成し、アセンブル・リストに付加する。

○　ラベルがオペランドとして用いられている箇所の、ステートメント番号を、ラベルに対応させ一覧表にした、クロス・リファレンス・リストを作成する。

2.1　オブジェクトの作成

オブジェクトのフォーマットは、Fig 2.1 のようなインテルフォーマットで作成される。

レコード長ＭＡＸは、３０バイト分であり、最後にチェック・サム（２の補数）と、ＣＲ（キャリッジ・リターン）、ＬＦ（ライン・フィード）が付加される。

チェックサムは、レコード長フィールドから、チェックサム・フィールドまでの総和の２の補数値が出される。

各データの水平パリティは、‘偶’である。

データのある場合、レコード・タイプは００、ファイルの終端（ＥＯＦ）の場合、タイプは、０１である。

0	:	レコード・マーク
1	(1)	レコード長フィールド ASCII HEX
2	(E)	
3	High	ロードアドレス・フィールド ASCII HEX
4		
5		
6	Low	
7	(0)	レコード・タイプ・フィールド データ ００ EOF ０１
8	(1)	
9		データ・フィールド ASS HEX
10		
9+(1E)×2		チェック・サム ASC HEX ２の補数値
9+(1E)×2+1		
10+(1E)×2	0D	CR
10+(1E)×2+1	0E	LF

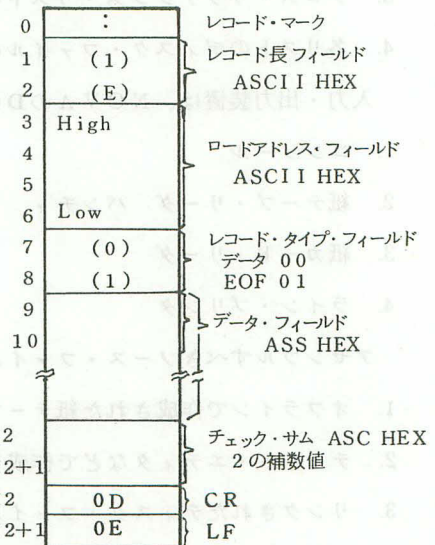



Fig 2.1

2.2 リストの作成

アセンブル・リストは、Fig 2．2のような形式で出される。

1行内に、エラー・コード、ステートメント番号、アドレスまたは、EQUで指定されたデータ、及びマシン・コード（最大4バイト分）の順で出力され、その後ソース行が出される。

```
 91  00DD                TBUFF:   DEFS     NTITL+1
 92  00EE                DATE:    DEFS     11
 93  00F9                PS0R:    DEFS     2
 94  00FB                BTAB:    DEFS     2
 95                      ;********* LTAB *********
 96  1900                LTAB     EQU      1900H
 97  3FFF                M16K     EQU      3FFFH
 98                      ;********* MAIN *********
 99                               ORG      100H
100  0100  1E00          MAIN:    LD       E,0             Fig 2.2
101  0102  21FF3F                 LD       HL,M16K
102  0105  220200                 LD       (MTAB),HL
103  0108  210019        MA02:    LD       HL,LTAB
104  010B  220000                 LD       (PTAB),HL
105  010E  1802                   JR       MA06
106  0110  1E40          CROSS:   LD       E,40H
```

番号　アドレス（データ）　マシン・コード　ソース行

エラー・コード桁

エラー・コードは行の先頭に出される。

ステートメント番号は、5桁の10進数で、ゼロサプレスされ、出力される。

アドレスは、4桁の16進数である。

マシン・コードは、最大4バイトまで同一行に出される。これを越えた分は、次の行に出される。

2.3 シンボル（ラベル）・テーブルの作成

指定（後述）がない限り、このシンボル・テーブルは自動的に、アセンブル・リストの後に付加され、出力される。

1桁に4個のシンボルとそれの置かれているアドレス、または等価なデータが出力される。

指定（後述）によって、シンボルは出現順、またはアルファベット順のいずれにでも配列できる。

2.4　クロス・リファレンス・リストの作成

シンボルがオペランドとして使用されている個所すべての、ステートメント番号が、各シンボルに対して付加される。

指定（後述）により、シンボルの配列を出現順、またはアルファベット順のいずれでも作成できる。

2.5　ディスク・ファイルの作成

上述の3種の出力（リスト、オブジェクトなど）は、直接各装置へ出力せず、一旦ディスク・ファイルとして作成しておくことができる。

このディスク・ファイルは、DOSの管理下で他のファイルと同様な扱いが可能である。

3.　アセンブラの書式

ソース・プログラムは、以下に述べるような書式（フォーマット）で作成する。

アセンブリ語として、オペコードはザイログ社オリジナルのニモニックを使用し、他に一般的なアセンブリ語を数種使用する。

a、マシン・コードに変換されるアセンブリ語

　　ＬＤ、ＡＤＤ、ＪＰなど６８種

b、その他のアセンブリ語（擬似命令語）

　　ＯＲＧ、ＥＮＤ、ＤＥＦＭなど７種

各文（ステートメント）は、１行単位で記述し、行終端は必ず、復帰（ＣＲ：キャリッジ・リターン）、または改行（ＬＦ：ライン・フィード）でなければならない。

＊　以下の説明の為に、ここで次のような記号を定めておく。

␣（ＳＰ）　；スペース（空白）を意味する。

⌣（ＨＴ）　；タブレーション（タブ）を意味する。

↲（ＣＲ）　；キャリッジ・リターン（復帰）を意味する。

≁（ＬＦ）　；ライン・フィード（改行）を意味する。

，（ＣＭ）　；コンマ（区切）を意味する。

ソース・プログラムで、最少限必要なアセンブラに対する命令語は、‘ＥＮＤ’であり、プログラムの最後に必ず書かねばならない。

ロードすべき、ロケーションの初めを指定するために、‘ＯＲＧ’命令が必要であるが、指定が無ければ、自動的に０（ゼロ）番地から始められる。

3.1　行の書式

１行内に、ラベル、オペコード（ニモニック）、オペランド、コメント文の順に書き、行終端はＣＲ（キャリッジ・リターン）または、ＬＦ（ライン・フィード）でなければならない。

行内の各要素は、全部必要な訳ではなく、次のような種々の行構成があり得る。

○　ラベルのみの行

○　オペコードのみの行

○　コメント文のみの行

○　オペコード、オペランドのみの行

○　そ　の　他

3.2　記　述　子

行内の各要素（文字列）を特徴付けるために、次のような各種の記述子（記号）が設けられている。

␣（SP）	；これの前後にある文字列を、分離する。
⌣（HT）	；　　　　　〃
，（CM）	；　　　　　〃
：（コロン）	；これの前にある文字列を、ラベルとする。
；（セミコロン）	；これの後にある文字列を、コメント文とする。
'（クォーティション）	；これらで挾まれた文字を、アスキーコードのまゝで扱う。
〈,〉（ブラケット）	；これらで挾まれた文字列を、数値として扱う。（但し、DEFM命令内のみ有効）
＋（プラス）	；これの後の文字列と等価な値を、前の文字列と等価な値に加える。
－（マイナス）	；これの後の文字列と等価な値を、前の文字列と等価な値から減じる。
＊（アスタリスク）	；これの後の文字列と等価な値を、前の文字列と等価な値に乗ずる。
／（スラッシュ）	；これの後の文字列と等価な値を、前の文字列と等価な値から除す。
＆（アンド）	；これの後の文字列と等価な値と、前の文字列と等価な値との論理積をとる。
！（感嘆符）	；これの後の文字列と等価な値と、前の文字列と等価な値との論理和をとる。

3.3　数値、式

オペランドとして、通常数値及び文字列を含む式が使える。

この内、数値には、次の４種が許される。

a、２進数　　　；数字列の後尾に‘B’を付す。

　　　　　　　（８桁まで、許される）

```
1011B
   ↑
```

b、8進数　　　　；数字列の後尾に、'O' または 'Q' を付す。（5桁まで、許される）

　　340　　　34Q
　　　↑　　　　↑

c、10進数　　　；数字列の後尾に、'D' を付すか、または、何も付さない。（5桁まで）

　　56D　　　56
　　　↑　　　　↑

d、16進数　　　；数字列の後尾に、'H' を付す。（4桁まで許される）数字列の先頭に'0'（ゼロ）を付すことが、望ましい。（先頭文字が、A～Fの場合、まずレジスタ名として、次にラベルとして調べられる。一致しなければ、16進数として、処理される）

　　78H　　　0ABH
　　　↑　　　↑　　↑

式は、演算記号で結ばれた文字列、数字群の一組であり、演算記号は、前述（下に再記）の6種が許される。

　＋、－、＊、／、&、!

演算は、行の左から順に行われる。括弧でくくることは、許されない。

e、数値式　　　　　23H－10＋3Q／4H＊8（＝38H）

f、文字（ラベル式）　LTAB＋BIAS&MASK

式内のSP（スペース）、HT（タブ）は無視される。

但し、CM（コンマ）は、式毎の分離を行う機能を有する。

数値、ラベル以外に数値と等価な記号として、次の2種がある。

g、ピリオド(.)　；その行に対応するロケーションの値（アドレス値）が代入される。

　　JP　　.＋5　　（＝JP　105H　アドレス：100番地）

h、ダラー($)　　；ピリオドに同じ。

　　JP　　$－3　　（＝JP　4FDH　アドレス：500番地）

オペコードが、多バイト命令である場合、その先頭バイトのアドレス値が採られる。

尚、オペランドとして、アスキー文字を使うことができる。

i、クォーティション(')　；これで挾まれた1文字のコードを、代入する。

　　LD　　A、'?'　（＝LD　A、3FH）

　　CP　　'<'＋2　（＝CP　3EH）

但し、クォーティション・マーク自身を挟むことは、許されない。

ＬＤ　Ａ、'''　（フォーマット・エラーとなる）

3.4　ラ　ベ　ル

ラベルは、文字、数字から成る６文字以内の文字列で書き、ロケーション、またはオペランドの値を充当させる。

先頭の文字に、数字は使えない。

ラベルは、その文字列の後尾に、':'（コロン）を付すか、行の先頭（第１コラム）から、書き始める。

␣ＡＢＣ：－－－

ＡＢＣ　　－－－

↑
行の先頭（第１コラム）

ラベルの文字数が、６文字を越える場合、越えた分については無視される。

異なるラベルを、同一ロケーションに設定したい場合、次のように書く。

Ｅ２００	ＡＢＣ：↲
Ｅ２００	ＥＦ：↲
Ｅ２００	ＪＫＬＭ：␣ＬＤ␣Ａ、Ｂ↲
↑ （アドレス）	（ラベル）

同一名のラベルを、複数個定義することは、許されない。

オペコードと同一名のラベルを使用してもよい。

使用できるラベルの総数は、メモリの残量によって決まる。

3.5　オペコード（ニモニック）

オペコードは、ザイログ社のニモニックを使用する。

データ転送	；ＬＤ、ＰＵＳＨ、ＰＯＰ
ブロック転送	；ＬＤＩ、ＬＤＩＲ、ＬＤＤ、ＬＤＤＲ
データ交換	；ＥＸ、ＥＸＸ
データ比較	；ＣＰ
ブロック探索	；ＣＰＩ、ＣＰＩＲ、ＣＰＤ、ＣＰＤＲ
算術演算	；ＡＤＤ、ＳＵＢ、ＳＢＣ、ＡＤＣ

論理演算　　　；ＡＮＤ、ＯＲ、ＸＯＲ

データ操作　　；ＩＮＣ、ＤＥＣ、ＤＡＡ、ＣＰＬ

ＮＥＧ、ＣＣＦ、ＳＣＦ

ＲＬＣＡ、ＲＬＡ、ＲＲＣＡ、ＲＲＡ

ＲＬＣ、ＲＬ、ＲＲＣ、ＲＲ、ＳＬＡ

ＳＲＡ、ＳＲＬ、ＲＬＤ、ＲＲＤ

ＢＩＴ、ＳＥＴ、ＲＥＳ

ＣＰＵ制御　　；ＮＯＰ、ＨＡＬＴ、ＤＩ、ＥＩ、ＩＭ*

ジャンプ　　　；ＪＰ、ＪＲ、ＤＪＮＺ、ＣＡＬＬ

ＲＥＴ、ＲＥＴＩ、ＲＥＴＮ、ＲＳＴ

入・出力　　　；ＩＮ、ＯＵＴ

ブロック入・出力　；ＩＮＩ、ＩＮＩＲ、ＩＮＤ、ＩＮＤＲ

ＯＵＴＩ、ＯＴＩＲ、ＯＵＴＤ

ＯＴＤＲ

*註）ＩＭ０、ＩＭ１、ＩＭ２　許される。

或は、０、１、２を、オペランドとしてもよい。

3.6　オペランド

オペランドとして、所定の文字、前述の数値、式、アスキー・コード、文字列などを使う。

ａ、所定の文字とは、次に挙げるレジスタ名、条件名など。

Ａ　　；アキュムレータ

Ｂ　　；汎用レジスタ

Ｃ　　；　〃

Ｄ　　；　〃

Ｅ　　；　〃

Ｈ　　；　〃

Ｌ　　；　〃

Ｆ　　；フラグ・レジスタ

ＡＦ　；レジスタ・ペア

ＢＣ　；　〃

ＤＥ　；　〃

ＨＬ　；レジスタ・ペア

ＩＸ　；インデクス・レジスタ

ＩＹ　；　〃

ＳＰ　；スタック・ポインタ

ＮＺ　；ノン・ゼロ

Ｚ　；ゼロ

ＮＣ　；ノン・キャリ

Ｃ　；キャリ

ＰＯ　；パリティ奇

ＰＥ　；パリティ偶

Ｐ　；正

Ｍ　；負

尚、間接指定に用いる括弧‘（）’、‘（ＩＸ±）’、‘（ＩＹ±）’などは、書式を変えてはいけないが、次のような書式は許される。

ＥＸ˙（ＳＰ）ＨＬ　　；＝ＥＸ　（ＳＰ）、ＨＬ

ＬＤ　Ａ␣␣（ＩＸ＋ｄ）　；＝ＬＤ　Ａ、（ＩＸ＋ｄ）

アスキー・コード、文字列をオペランドとして使う例については、アセンブラ命令の項で説明する。

3.7　アセンブラ命令

本来の命令語以外に、アセンブラに対する擬似命令（制御）語がある。

アセンブラ命令には、次の８種があり、一部省略形が許される。

ＴＩＴＬＥ　; Title

ＯＲＧ　; Origin

ＥＮＤ　; End of Source

ＥＱＵ　; Equalize

ＤＥＦＳ　; Define Storage area

（ＤＳ　;　〃　）

ＤＥＦＢ　; Define Byte space

（ＤＢ　;　〃　）

DEFW　　　; Define Word space

（DW　　　;　　　〃　　　）

DEFM　　　; Define Massage strings

（DM　　　;　　　〃　　　）

各々について、詳細に説明する。尚、書式中の⌣は、␣でもよい。

a、ORG

書式　　⌣ORG⌣ABC↲

意味　　以降のプログラムを、ロケーションABCから、ロードする。

条件　　ABCは、数値、または式であり、ラベルを含む場合、そのラベルは、この行以前で定義されている必要がある。

b、END

書式　　⌣END↲

意味　　ソース・プログラムの終端とする。

条件　　ラベルを付すことができるが、オペランドを付けることは許されない。

c、EQU

書式　　XYZ：⌣EQU⌣ABC↲

意味　　ラベルXYZの値として、ABCを与える。

条件　　ABCは、数値、または式であり、ラベルを含む場合、そのラベルは前もって定義されていなければならない。

d、DEFS

書式　　⌣DEFS⌣ABC↲

　　　　⌣DS⌣ABC↲

意味　　ストレージ・エリアを、ABCバイト確保する。

条件　　ABCは、EQUのと同様で、ラベルを含む場合は、そのラベルが、前もって定義されていること。

e、DEFB

書式　　⌣DEFB⌣ABC↲

　　　　⌣DB⌣ABC↲

意味　　ABCの値の下位1バイトの、マシン・コードを出力する。

条件　　ABCは、数値、または式で、カンマで区切って、複数組並べて書いてもよい。

f、DEFW

書式　　　␣DEFW␣ABC↲

　　　　　␣DW␣ABC↲

意味　　　ABCの値2バイトの、マシン・コードを出力する。

条件　　　ABCは、数値、または式で、カンマで区切って、複数組並べて書いてもよい。

g、DEFM

書式1.　␣DEFM␣／ABC／↲

　　　　␣DM␣／ABC／↲

意味　　　アスキー・コードA、B、Cを、この順序で出力する。

条件　　　文字列を同一のターミネータで挟む。このターミネータは、制御コード以外なら、何でもよい。

　　　　　ターミネータのコードは出力されない。

書式2.　␣DEFM␣／ABC／〈03FH、6〉

　　　　␣DM␣／A／〈5〉／B／〈0FH〉

意味　　　アングル・ブラケットで挟まれた数値、式の値はそのまゝ、数値として出力する。

　　　　　ターミネータの間に挟まれたアングル・ブラケットは、文字コードと見なされるので、注意すること。

h、TITLE

書式　　　␣TITLE␣ABC↲

意味　　　文字列ABCを、ソース・プログラムのタイトルとする。

条件　　　文字列の長さは、10文字迄で、これを越える分は無視される。

　　　　　タイトルは、リスト類の打出しの際にのみ有効であり、ファイル名とは無関係である。

4. アセンブル出力

アセンブル出力は、先述したように、４種の形がある。

この内、アセンブル・リストは、ソースに種々の情報が付加されたものであり、プログラムの確認、修正、改良を容易にするのに役立つ。

4.1 リストの内容

リストの先頭には、リアルタイム・クロックが装備されていれば、年月日の付されたメッセージが出される。

タイトル行は、パス１で、行全部が打ち出されるが、タイトル文は、文字・数字の初めの１０文字のみが有効である。

エラーのある行については、後述する。

ステートメント番号は、すべてのソース行について付される。但し、アセンブラによって、余分に作成される行については、付されない。

ロケーション（アドレス）は、‘００００’～‘ＦＦＦＦ’で表わされる。

‘ＦＦＦＦ’を越えて、マシン・コードを発生させねばならない場合、アセンブラは、エラーを出す。

マシン・コードは、１行内に最大４バイト（８コード）出し、それを越える分については、次行に出す。

ＥＱＵ命令のオペランドに対応する値は、マシン・コードのコラムに出されるが、実際のマシン・コードは出力しない。

エラーのある命令行において、できる限りのコードが作成され、作成し得ない箇所のみ、ＮＯＰ（００）コードが出される。

4.2 エラー・コード

エラーのある行の先頭に、１及至２文字のコードが出される。

エラー・コードは、次の９種で、他にコーション・コード１種がある。

ａ、コードＢ　　；バッド・キャラクタ

不適当なキャラクタを使用している。

b、コードF　　;フォーマット・エラー

書式が、正しくない。

c、コードJ　　;ジャンプ・エラー

相対ジャンプで、ジャンプ先が遠過ぎる。

e、コードL　　;ラベル・エラー

ラベルが、不適当である。

f、コードM　　;マルチ・ラベル・エラー

同一名のラベルが、複数個ある。

g、コードP　　;フェーズ・エラー

ラベルの値が、パス毎に異なっている。

h、コードR　　;リファレンス・エラー

フェーズ・エラーを生じたラベル、または、複数個あるラベルを参照した。

i、コードU　　;アン・デファイン・エラー

未定義のラベルを参照した。

j、コードV　　;オーバー・エラー

値が、許される範囲を越えた。

コーション・コードとして、次のものがある。

k、コード!　　;コーション

ラベルの文字数が、6文字を越えた。

許されるが、7文字以降は無視されるので、注意すること。

5. 操　　作

クロス・アセンブラは、通常リロケータブル・バイナリの形で、提供されるので、これを実行させる為には、セーブ・ファイルにしておかねばならない。

紙テープで提供された場合のロード、及び実行について述べる。

1. テープを、リーダにセットする。
2. コンソールから、次のようにキー・インする。

RLDR␣Z80A／S␣＄PTR↲

3. CLIから、'R' が打ち出された時点から実行可能となる。

5.1 アセンブル・コマンド

ソースをアセンブルする為に、次のような形式で、コンソールから、コマンドをキー・インする。

Z80A（／A／N）␣input－ficl␣（output－file／L／X／B）↲

アセンブラ名　グローバル・スイッチ　入力ソース名　出力ファイル名　ローカル・スイッチ

（　）内は省略可。複数指定可。

アセンブラ名を、変更（RENAME）してはいけない。

グローバル・スイッチ

／A　；シンボル（ラベル）・テーブル、及びクロス・リファレンス・リストのラベル配列を、アルファベット（ABC）順にする。

このスイッチを付けなければ、ラベル配列は、出現順となる。

／N　；シンボル・テーブルを、リスティングしない。

このスイッチがなければ、アセンブル・リストを要求した時、自動的にシンボル・テーブルが付加される。

入力ソース名は、アセンブルしたいソースのファイル名である。このファイル名には、スイッチを付加してはいけない。

出力ファイルは、適当なディスク・ファイル名、または出力デバイス名である。

例）　ABC、＄LPT、＄PTP

このファイル名は、入出力デバイスでなければ、入力ファイル名と同一名でもよい。その場合、出力ファイル名は、入力ファイル名に、拡張子が付された名となる。

拡張子は、ローカル・スイッチの指定により異なる。

ローカル・スイッチ

／Ｌ　；このファイルに、アセンブル・リストを出力する。

ディスク・ファイルの場合には、このファイル名に、拡張子 '．ＬＳ' が付加される。

／Ｂ　；このファイルに、オブジェクト（アブソリュート・バイナリ）を出力する。

ディスク・ファイルの場合には、このファイル名に、拡張子 '．ＡＢ' が付加される。

／Ｘ　；このファイルに、クロス・リファレンス・リストを出力する。

ディスク・ファイルの場合には、このファイル名に、拡張子 '．ＸＲ' が付加される。

出力ファイルには、必ずいずれかのローカル・スイッチを付す必要がある。

出力ファイルが、指定されなくても、アセンブルを行い、エラー箇所のみ、コンソールに打ち出させることができる。

5.2　アセンブルの実行

ＤＯＳのＣＬＩ下で、先述のコマンドを、コンソールからキーインする。

例１）　ディスク・ファイル 'ＡＢＣ' の、アセンブルのみ行わせる。（エラーの有無を調べたい場合）

Ｚ８０Ａ␣ＡＢＣ↲

例２）　ディスク・ファイル 'ＡＢＣ' を、アセンブルし、ライン・プリンタに、リストを出力させる。

Ｚ８０Ａ␣ＡＢＣ␣＄ＬＰＴ／Ｌ↲

例３）　紙テープ・ソースを、アセンブルし、紙テープのオブジェクトを、出力させる。

Ｚ８０Ａ␣＄ＰＴＲ␣＄ＰＴＰ／Ｂ↲

例４）　ディスク・ファイル 'ＡＢＣ' を、アセンブルし、ライン・プリンタにリストを、ディスクへクロス・リファレンス・リストと、オブジェクトを出力させる。

シンボル・テーブルは、付加しない。

Ｚ８０Ａ／Ｎ␣ＡＢＣ␣ＡＢＣ／Ｘ／Ｂ␣＄ＬＰＴ／Ｌ↲

（ディスク・ファイル {ＡＢＣ．ＸＲ / ＡＢＣ．ＡＢ} が、作成される。）

同じローカル・スイッチを、異なるファイルに付した場合、先に付加されたファイルのみが、有効となる。

a、アセンブルの中止、中断

コマンドが、正しくキー・インされなかった場合、'？'を打ち出して、CLIに戻る。

入出力に関する制御などは、DOSのシステム・コールを利用しているので、人出力異常時には、DOSがメッセージを出す。

オペレータの意志で、アセンブルを中断させる場合、通常は、次のような操作で行う。

＊　CTRLキーとAキーを同時に押す。

これによって、アセンブルは中止され、CLIに戻る。

この時、スイッチ'X'で、クロス・リファレンス・リストが要求されている場合、次のディスク・ファイルが、作成されたままになることがある。

XRTMP

残しておいても、差支えはないが、消去しておくとよい。

DELETE␣XRTMP↲

アセンブルの途中で、ラベル数が規定値を越えると、次のメッセージを打ち出して、アセンブルを中止し、CLIに戻る。

'!!LABEL　SAVING　AREA　OVER−FLOWED!!'

b、アセンブルの終了

アセンブル・コマンドで指定された項目が、すべて遂行された時点で、アセンブルが終了し、CLIに制御が戻る。

Z-80
テキストエディタマニュアル

9

目　　　　　次

ＳＭ－Ｂ－８０Ｄ　テキストエディタ　ＥＤＩＴ

1. 概　要

ＳＭ－Ｂ－８０　Ｄテキストエディタは、ＳＭ－Ｂ－８０　Ｄモニタ（ＬＨ－８Ｓ０３）と共に用い、ＡＳＣＩＩコードの文字列を作成、修正、編集することができる。

このテキストエディタを用いれば様々な文字列を作成でき、特にマイクロコンピューター等でのアセンブラソースの作成、及びアセンブルした後でのデバッグの為の修正に適している。ソース紙テープ、あるいは、キーボードによってソースを入力し、ソースの文字の削除、追加、変更等が、文字単位あるいは行単位で簡単に行なえ、出力として、ソース紙テープを新しくパンチすることができる。

2. エディトバッファ

1）エディトバッファ　　ソースプログラムのメモリ領域とキーボードよりの入力のメモリ領域として使用される。ソースのメモリ領域とキーボードよりの入力のメモリ領域は共存しており、その占有する割合は可変となっている。エディトバッファは、常に有効な活用を行なっている。

2）バッファポインタ　　ソースプログラムの文字列を扱っていく場合、１文字を１単位として処理を行なう。その場合に、扱っている文字の位置を示す必要があり、それをバッファポインタとして以下で述べるコマンドの実行の基準となる。エディタバッファ内でバッファポインタはソースプログラムのいかなる文字に動かすことも可能である。

3. 文字の使用

1）ＣＲ（Carrige　Return）　　エディタバッファ内に於ては、ＣＲが行の終りとしての意味をもち、ＣＲごとに行単位が構成される。エディタの制御の中では、ＣＲは常にＬＦ（Line　Feed）の機能を合せて行なう。（ＣＲがキーボード入力されると、ＣＲとＬＦが一緒に印字される。）

2）ＬＦ（Line　Feed）　　エディタバッファ内には入力されないようになっている。従って、紙テープ入力及びキーボード入力でＬＦはすべて読みとばされる。

3）ＴＡＢ（Tab）　テキストエディタは、ＴＡＢキーあるいはＣＴＲＬ／Ｉキーによってｔａｂ文字が入力される。これは、バッファ内では、９Ｈとして書きこまれているが、印字さ

れる時には、水平ＴＡＢとして、８文字毎に文字をそろえる役目をもつ。

4）ＥＯＴ（End　of　Tape）　紙テープの終了という意味をもつ。

5）ＦＦ（Form　Feed）　紙テープの一つの単位を示す。従ってＦＦあるいはＥＯＴによって紙テープ入力は停止する。又、この２つのコードは、バッファ内には入力されないで、読みとばされる。

4. テキストエディタの動作の開始、終了

1）エディタ動作の開始

以下のキーボード入力によってエディタの実行が始まる。

＊Ｅ；Ｇ↲　　＊は、モニターのコマンド待ちの状態を示す。

↲は、ＣＲを示す。（ＰＲＯＭバージョンの場合、．ＥＤ；Ｇ↲）

この入力に対してエディタは、実行の開始を示すために以下の文を印字する。

↲
ＥＤＩＴ␣Ｖｘ．ｙ↲　　ｘ．ｙは、エディタの仕様に対応する。
＊

＊は、エディタ動作での、キーボードでのコマンド入力の準備完了及び、前のコマンドの実行終了を示す。従って＊が印字されたら、コマンドを入力していくことができる。

2）エディタコマンドの入力

エディタコマンドはキーボードで入力する。そしてコマンド文は一部を除いてＥＳＣキーによって区切られ、ＥＳＣキー２回続けてうつことにより、コマンドは実行される。尚ＥＳＣキーに対して、'＄'が印字される。そして、コマンドが実行され終了したことに対して'＊'が印字され、新しいコマンドを待つ。

3）エディタ動作の終了

エディタは、以下の２つのコマンドのどちらかでその動作を終了する。

＊Ｈ＄＄　　モニターへ制御をうつす

＊Ｘ＄＄　　アセンブラに制御をうつす。（アセンブラについては、別資料参照）

5. 入出力

1）入　力　　ソースの入力には、次の２種類がある。

a、テープ入力　　Ａコマンド

紙テープからテープリーダーによってエディトバッファに直接入力される。入力の終了

は、ＦＦあるいはＥＯＴによって判別される。従って、紙テープ入力の場合最後には常に

ＦＦかＥＯＴが必要である。

ｂ、キーボード入力　Ｉコマンド

Ｉコマンドによってキーボードからの入力が行なわれる。エディタバッファへはＩ以下の文字が入力される。

２）出　力　出力にはソースの内容を印字する方法と、紙テープ出力を行なう方法がある。

ａ、テープ出力　Ｅ、Ｏ、Ｐ、Ｙコマンド

ソースの内容をＡＳＣＩＩコードによってパンチする。パリティは、偶数である。

ｂ、タイプ出力　Ｔ、Ｗコマンド

ソースの内容を印字する。

6. コマンドの説明

SM−B−８０ Ｄのテキストエディタのコマンドは、すべてアルファベット１文字から成立っており、そのうちいくつかには数字あるいは負記号が添えられる。

以下に述べるコマンドの説明において、まずエディタバッファ内には、下の様なプログラムが格納されているとする。又、'＄'はＥＳＣキーのエコーバックである。

```
;(TAB)        ***   EXAMPLE   ***↲
(TAB)         XOR(TAB)   A↲
(TAB)         LD(TAB)    B,50↲
(TAB)         LD(TAB)    HL,BUFFER↲
MOV:(TAB)     LD(TAB)    (HL),A↲
(TAB)         INC(TAB)   HL↲
(TAB)         DEC(TAB)   B↲
(TAB)         JP(TAB)    NZ,MOV↲
(TAB)         END↲
```

注：↲はＣＲ（０ＤＨ）を示す。

(TAB)はＴＡＢコード（０９Ｈ）を示す。

▽はバッファポインタ（ＢＰ）を示す。

１）Ａ　（Append）

Form　　Ａ＄

このコマンドで紙テープ入力を指定し、テープ入力をする。読みこまれた入力はバッ

内の既入力の末尾からバッファに書きこまれていく。紙テープの終了は、紙テープにパンチされたＥＯＴコードあるいはＦＦコードによって行なわれる。ＥＯＴとＦＦコードは、バッファ内には書きこまれないが、そのどちらかのコードで終了したかは保存されている。又、ＮＵＬＬ（０Ｈ）、ＤＥＬ（７ＦＨ）は読みとばされる。

〈例〉

現在下の様な位置にＢＰ（バッファポインタ）があるとする。

```
(TAB)          DEC(TAB)       B↲
▽
(TAB)          JP(TAB)        NZ，MOV↲
(TAB)          END↲
```

この時、下の様な紙テープをセットして、Ａコマンドをキーボード入力してやる。

紙テープの内容

←

NULL	NULL	NULL	;	TAB	*	*	SPACE	E	D	I	T	SPACE	E	N	D	SPACE	*	*	CR	LF	EOT	NULL	NULL	NULL	NULL	NULL	NULL	NULL

キーボード入力の内容

```
A (ESC)(ESC)
```

タイプされる内容

```
〔*〕A$$
 ↑   ↑↑
 │   └┴── ESCキーのエコーバック
 └──── テキストエディタがコマンド待ちであることを表わす文字
```

以上の操作により、テキストエディタは紙テープの読み込みに入りＥＯＴコードを読んだ個所で紙テープ入力は終了する。

Ａコマンド終了後バッファには、次の様に入力されている。又、バッファポインタ（ＢＰ）は、位置を変更していない。

```
(TAB)          DEC(TAB)       B↲
▽
(TAB)          JP(TAB)        NZ，MOV↲
(TAB)          END
;(TAB)         ** EDIT        END **↲
```

2） B (Begin)

Form　B＄，nB＄　　0≦n≦65535

このコマンドは、‘B＄’でバッファポインタをソースプログラムの先頭へ位置させるものである。又、‘n＄’は、先頭より数えてn行目の先頭へバッファポインタを位置させるコマンドである。

〈例〉

現在、下の様にバッファポインタがあるとする。

;(TAB)　　　＊＊＊　EXAMPLE　＊＊＊↵
(TAB)　　　XOR(TAB)　　A↵
▽
(TAB)　　　LD(TAB)　　A，50↵

この時、下の様にコマンドを入力する。

〔＊〕B＄＄

バッファポインタは▽1の位置にある。

▽1 ;(TAB)　　　＊＊＊　EXAMPLE　＊＊＊↵
(TAB)　　　XOR(TAB)　　A↵
(TAB)　　　LD(TAB)　　A，50↵
▽2 (TAB)　　　LD(TAB)　　HL，BUFFER

次に、下のコマンドを入力すると、バッファポインタは▽2の位置にくる。

〔＊〕4B＄＄

cf.1.　B＄と0B＄と1B＄は、同じ命令を意味している。

2.　Bコマンドの次の＄(ESCキー)は、省略して続けて新たにコマンドを入力することも可能である。下の例は同じ命令である。

〈例〉

〔＊〕B＄A＄＄

〔＊〕BA＄＄

3） C (Change)

Form　　C strings 1＄　strings 2＄

文字列strings 1を文字列strings 2に変更する。変更するのは、バッファポインタより後のプログラムで、最も近い指定された文字列である。コマンドの実行後、バッファポインタは変更した文字列の次となる。strings 2がない場合は、strings 1を削除する。

〈例〉

現在下の様なプログラムの位置にバッファポインタがあるとする。

```
▽
(TAB)          LD(TAB)     HL，BUFFER↵
MOV：(TAB)     LD(TAB)     (HL)，A↵
```

次のコマンドを入力する。

〔＊〕CBUFFER＄SPACE＄＄

結果は下の様になる

```
(TAB)          LD(TAB)     HL，SPACE↵
MOV：(TAB)     LD(TAB)     (HL)，A↵
```

次に下のコマンドを入力する

〔＊〕CMO＄＄

結果は下の様になる。

```
(TAB)          LD(TAB)     HL，SPACE↵
▽
V：(TAB)       LD(TAB)     (HL)，A↵
```

4）　D（Delete）

Form　D＄，nD＄　　0≦n≦65535

文字を現在のバッファポインタからn文字削除する。D＄と1D＄は同じである。

〈例〉

```
▽
V：(TAB)       LD(TAB)     (HL)，A↵
(TAB)          INC(TAB)    HL↵
```

上の状態で次のコマンドを入力する。

〔＊〕3D＄＄

次の様になる。

```
▽
LD(TAB)        (HL)，A↵
(TAB)          INC(TAB)    HL↵
```

5） E（End）

Form　　E＄

エディタバッファ内の内容を最初からすべてパンチする。そして、入力においてAコマンドが実行されていて、かつその終了状態が、エディタバッファのフル（Buffer full）だった場合、テキストエディタは、紙テープを読みそれを複写していき、EOTあるいはFFを複写して終了する。その後256個の空白（Null）をパンチする。

この動作終了後、バッファ内の内容は初期化される。

6） F（Feed）

Form　　F＄

256個の空白をパンチするコマンドである。これによって紙テープのリーダーやトレーラーを作成することができる。

〈例〉

エディタの編集終了で最初に512個の空白（Null）を入れて、エディタバッファの内容をすべてパンチし、エディタ動作を終了したい場合、下の様にコマンドを入力する。

〔✳〕F＄F＄E＄＄

7） G（Gain）

Form　　G＄

エディトバッファ領域の上限を指定するコマンドである。このコマンドにより、RAMの増設等によるエディトバッファ領域の変更が可能となる。この変更は、制御がモニターに戻るまで有効である。

又、このコマンドは、アセンブラとの共通コマンドとなっており、エディタとアセンブラでのみ制御されている場合には、どちらか一方で指定するだけで十分である。

〈例〉

〔✳〕G＄＄〔↲〕

8) H (Home)

Form　H $

テキストエディタの動作が終了し、制御をモニターへうつす。

このコマンドによって下の様に印字されモニター動作が始まる。

〈例〉

```
〔*〕H $ $〔↲〕 ←―――― このコマンドでエディタからモニタへ制御がうつる
〔↲〕
〔 SM-B-80D　V 1 . 1 〕←―― モニタ動作による印字
〔*〕←―――――――――――― モニタコマンド待ち
```

9) I (Insert)

Form　I strings　$

このコマンドはＩに続く文字列 strings をエデットバッファ内のソースプログラムに挿入するものである。挿入する位置は現在のバッファポインタの示す個所である。このコマンドの終了後、バッファポインタは挿入文字列の次に位置する。

〈例〉

```
                                ▽
(TAB)      L D(TAB)      B , 5 0↲
(TAB)      L D(TAB)      HL , BUFFER↲
```

以上の様な状態の時、次のようなＩコマンドを入力するとその結果は下の様になる。

```
〔*〕I 4 8 H↲
(TAB)      L D           C , $ $
```

〈結果〉

```
(TAB)      L D(TAB)      B , 4 8 H↲
                          ▽
(TAB)      L D(TAB)      C , 5 0↲
(TAB)      L D(TAB)      HL , BUFFER↲
```

次にエディタバッファの最初にＩコマンドで文字列を挿入するコマンドは次の様になる。

```
〔*〕B $ I(TAB)        **START**↲
$ $
```

これによって次の様に変更される

```
(TAB)                  **START**↲
▽;(TAB)                ***EXAMPLE***↲
(TAB)                  XOR     A↲
```

cf.Iコマンド入力の場合、ＦＦ（０ＣＨ）、ＥＯＴ（０４Ｈ）、ＮＵＬＬ（０Ｈ）、ＤＥＬ（７ＦＨ）、ＥＳＣ（１ＢＨ）、ＬＦ（９Ｈ）は、バッファ内には挿入されることはない。

エディタ動作では、上記のＦＦ、ＥＯＴ、ＮＵＬＬ、ＬＦがキー入力されても読みとばされる。

10）　Ｋ（Kill）

Form　Ｋ＄，ｎＫ＄　　０≦ｎ≦６５５３５

バッファポインタからｎ行文字列を削除する。Ｋ＄と１Ｋ＄は同じである。

〈例〉

以上の状態の時、次のコマンドを入力する。

〔＊〕２Ｋ＄＄

その結果は次の様になる。

11）　Ｌ（Line）

Form　Ｌ＄，ｎＬ＄　　－６５５３５≦ｎ≦６５５３５

バッファポインタを現在の位置から、ｎ行目の先頭に移動する。

Ｌ＄、０Ｌ＄、－０Ｌ＄は、バッファポインタを現在の行の先頭に移動するコマンドとなる。－Ｌ＄と－１Ｌ＄は同じコマンドである。

〈例〉

以上の様な状態であるとする。そして次のコマンドを入力するとバッファポインタは、次の様になる。

cf．コマンドが連続する時には、ESCキー（＄）を省略することができる。

12)　M（More）

Form　　M＄，nM＄　　－65535≦n≦65535

バッファポインタを現在の位置よりn文字移動するコマンドである。M＄と1M＄、－M＄と－1M＄は同じコマンドを意味する。

〈例〉

上のLコマンドの③にバッファポインタが位置する場合、次の様なコマンドを入力する。

〔＊〕－4M＄＄ ———— ④

〔＊〕10M＄＄ ———— ⑤

〔＊〕M＄＄ ———— ⑥

〈結果〉

MOV：(TAB)	LD (TAB)	(HL④)，A↲
(TAB)	INC (TAB)	⑤⑥HL↲

cf．コマンド連続する時は、ESCキー（＄）を省略することもできる。又、CR、TABは1文字として数えられる。

13)　N（Next）

Form　　N＄

エディトバッファ内をすべてパンチして出力し、バッファ内の内容を空にする。続いて紙

テープを読み込みエディトバッファに入力する。テープ入力は、ＦＦコード（０ＣＨ）、ＥＯＴコード（０４Ｈ）で終了する。バッファポインタは、エディトバッファの先頭に位置する。

14）　Ｏ（Out）

　　Form　Ｏ＄

２５６個の空白を紙テープにパンチする。続いてエディトバッファ内の内容を最初からパンチし、それが終了後ＥＯＴコード（０４Ｈ）をパンチする。続いて２５６個の空白をパンチする。

このコマンドでは、エディタバッファの内容及びバッファポインタは変化しない。

通常、このコマンドで紙テープの作成をすれば非常に有効である。

15）　Ｐ（Punch）

　　Form　Ｐ＄，ｎＰ＄　　$0 \leqq n \leqq 65535$

現行のバッファポインタよりｎ行文字列をパンチする。この場合、ＣＲはＣＲとＬＦをあわせてパンチされ、又ＴＡＢは０９Ｈとパンチされる。

Ｐだけでのコマンド（Ｐ＄）は、バッファポインタをそのままで、エディタバッファ内の内容を始めからパンチするコマンドとなる。

〈例〉

以上の時、下の様なコマンドを入力する。

〔＊〕Ｆ＄３Ｐ＄＄

〈結果〉

〔*〕P $ F $ $

16） Q （Quit）

Form　　Q $

エディトバッファの内容をすべて空にして、初期状態にもどすコマンドである。このコマンドの後に他のコマンドを続けて入力しても実行されない。

17） R （Rest）

Form　　R $

エディタ動作で現在使っていないエディトバッファ領域の大きさを示すコマンドである。このコマンドが入力されると次の様に出力される。

FREE CORE: 8000BYTES↲

これによって、8,000文字分文字列を挿入することができることがわかる。

18） S （Search）

Form　　S strings $

文字列 strings を現在のバッファポインタより捜していくコマンドである。バッファポインタは、捜した strings の次に位置する。

＜例＞

```
; (TAB)          ***  EXAMPLE  ***↲
▽
(TAB)            XOR (TAB)     A↲
(TAB)            LD (TAB)      B，50↲
(TAB)            LD (TAB)      HL，BUFFER↲
MOV：(TAB)       LD (TAB)      (HL)，A↲
```

上の状態の時、次の様なコマンドを入力する。

〔*〕SLD$ $

バッファポインタの位置は次の様になる。

```
                   ▽
(TAB)            LD (TAB)      B，50↲
(TAB)            LD (TAB)      HL，BUFFER↲
```

もし、捜す文字列が見つからない場合は、サーチエラーシンボルがタイプされる。この場合、バッファポインタは移動しない。

〈例〉

〔＊〕ＳＡＢＣ＄＄

!! ＡＢＣ ——捜した文字列

!! —— サーチエラーを示す

19） Ｔ（Type）

Form　　Ｔ＄，ｎＴ＄　　0≦ｎ≦６５５３５

現在のバッファポインタよりｎ行タイプする。Ｔ＄と１Ｔ＄は同じコマンドである。バッファポインタは移動しない。

〈例〉

```
; (TAB)          ▽***  EXAMPLE  ***↲
(TAB)            XOR (TAB)     A↲
(TAB)            LD (TAB)      B，50↲
```

次のコマンドを入力する。

〔＊〕２Ｔ＄＄ ———— 1

〔＊〕１Ｌ＄Ｔ＄＄ ———— 2

〈結果１〉

```
***  EXAMPLE  ***
          XOR          A
* ——コマンド待ち
```

〈結果２〉

```
          XOR          A
* ——コマンド待ち
```

このコマンドではＥＳＣキー（１ＢＨ）'＄'は省略することもできる。次の２つのコマンドは同じである。

〔＊〕５Ｔ＄３Ｌ＄＄

〔＊〕５Ｔ３Ｌ＄＄

20） U (Unit)

Form　　U＄

ＡＳＣＩＩコードの０ＣＨ (Form Feed) を紙テープにパンチする。

〈例〉

〔＊〕１Ｐ＄Ｕ＄１Ｌ＄１Ｐ＄＄

U＄（これによって区切りをつけることができる。）

21） V (Volume)

Form　　V＄

ＡＳＣＩＩコードの０４Ｈ (ＥＯＴ ; End of Text) を紙テープにパンチする。

22） W (Write)

Form　　W＄

エディトバッファの内容をすべて最初からタイプする。この場合、バッファポインタの位置は移動しない。

23） X (Cross)

Form　　X＄

制御をアセンブラに移す。この時、エディトバッファの内容は現状のままである。（エディタ単独の場合はこのコマンドは使用出来ない。）

24） Y (Yield)

Form　　Y＄

２５６個の空白を紙テープにパンチする。続いてエディトバッファの内容を最初からパンチし、それが終了後ＦＦコード（０ＣＨ）をパンチし、その次にＥＯＴコード（０４Ｈ）をパンチする。そして、２５６個の空白をパンチする。

このコマンドでは、エディトバッファの内容およびバッファポインタは変化しない。

サブルーチン等の紙テープパンチにこのコマンドを用いれば非常に有効である。

25） Z

Form　　Z＄，　nZ＄

バッファポインタをソースプログラムの最後に移動させる。また、'nZ$' は、バッファポインタをソースプログラムの最後から数えてn行目の先頭に移動させるコマンドである。

ESCキーは省略することができる。

Z$と0Z$は同じ命令である。また、nZ$はZ$－nL$と同じコマンドである。

7. エラー時等の処理

1）Aコマンド実行中にエディトバッファがいっぱいになった時

紙テープの読み込みを中止し、次のコメントをタイプしコマンド待ちの状態となる。

```
〔*〕A$$
BUFFER FULL!
〔*〕
```

この後、キーボードからコマンドを16文字まで入力できる。

2）キーボードからの入力の時、エディトバッファがいっぱいになった時

バッファがいっぱいであるコメントをタイプし、さらに、入力していったコマンドの末尾より16文字を削除し、その2行前よりコマンドをタイプしていく。従って、この後続けてキーボード入力をすればよい。

〈例〉

```
〔*〕I;   INITIAL  CONDITION↲
;  A=0,B=0,C=1,D=3↲
;            PROGAM   START↲
ABC:       LD
             └──────── ここでバッファがいっぱいになる。
```

```
BUFFER  FULL!
;  A=0,B=0,C=1,D=3
;          PROG
```

以上の様にタイプする。この状態は、キーボード入力で入力されてきたと考えてよい。従って、この状態に続けてコマンドを適当に入力する。あるいは、入力したコマンドを削除することができる。

バッファが一杯でコマンドが入力できない状態の時、コマンドを入力中に、バッファが一杯となるたびに、コマンドを1文字ずつ多く入力できるようになる。但し、これはコマンド

が４文字になるとこの作業は終了する。この時、バッファの最後の文字から１文字ずつ削除される。

3） コマンド文エラーの時

＄＄でコマンドが終了し、その入力方法にエラーがあった場合には、次の様に？？を先頭後にエラーコマンドをタイプする。

〈例〉

〔＊〕DABC＄＄

？？DABC ―――――― DABCというコマンドがわからない。

また、エラーの文字列が１６文字列以上の時は、先頭から１６文字のみタイプする。そして、次の様に訊ねてくる。

INSERT？（Y：YES）

この時もしIコマンドのつもりで、最初にIを入力することを忘れた場合なら、キーボードの‘Y’を入力する。そうすれば、そのエラーコマンドはIコマンドと同じ様に動作する。また、他のエラーの場合は、その他のキー入力を行なえば、コマンドは削除され新らたにコマンド待ちの状態になる。

4） 文字列の見つからない時

Sコマンド又はCコマンドで指定された文字列が見つからない時、次の様に！！を先頭にタイプして見つからない文字列をタイプする。

〈例〉

〔＊〕SABC＄＄

！！ABC

〔＊〕CABC＄DEF＄＄

！！ABC＄DEF

この場合、バッファポインタは前の位置から移動しない。

また、文字列が１６文字をこえてしかもそれがみつからない時には、コマンド文エラーと同じ様に‘INSERT？’と訊ねてくる。この時、Yをキー入力すれば、S或はCから文字列は挿入される。

8. キー入力の意味

1） DELキー（＝RUB OUTキー）

前に印字した文字を１文字削除する。そして、削除された文字は‘＼’によってはさまれる。

〈例〉

```
                                 ┌──────1回目のDELキー
                                 │  ┌────────2回目のDELキー
〔*〕I; ** APPLY\YL\END **
                         └──────── DELした代り入力する文字
$$
```

上の様に入力した時には、エディトバッファ内には次の様になる。

; ** APPEND **↲

つまり、'LY' を削除している。

2) CRキー

このキーを入力することによって、CRとLFが同時に印字される。

3) LFキー

このキーは無視される。

4) ESCキー

このキーの入力に対し、'$' 文字が印字される。

5) TABキー（Control Iキー）

8文字単位で文字をそろえる役目をもつ。

6) Control Xキー

入力しているコマンドを一行削除する。

7) Control Zキー

入力しているコマンドをすべて削除し、コマンド待ちの状態となる。

8) Control Cキー

コマンドの実行を中断し、新しくコマンド待ちの状態となる。

但しこのコマンドは、パンチ或はタイプの時にのみ有効である。

9) その他コントロールキー

コントロールキーを押して入力したことを示すために、'↑' が印字され次にコントロールキーを押さない状態での文字が印字される。

〈例〉

↑A　コントロールAキー

↑B　コントロールBキー

Control Cキーを入力した場合、コマンドが中断されると次の様に印字する。

```
(TAB)  LD (TAB)  A,(HL)
                   ↑──────── ここでControl Cの入力を行なう
(TAB)  LD (TAB)  A,
CANCEL!
*←───────コマンド待ち
```

エディタコマンド一覧表			（($)は省略可、(−)nは負も可を示す。）
No.	入力形式	名称	説明
1	A $	Append	紙テープ入力を、エディトバッファの既入力の末尾から行なう。
2	B $ nB ($)	Begin	バッファポインタを先頭よりn行目の先頭へ移動させる。
3	C strings1$ strigs2 $	Change	strings 1をstrings 2に変更する。
4	D $ nD $	Delete	現在のバッファポインタからn字削除する。
5	E $	End	エディトバッファ内をすべてパンチし、バッファの内容を初期化する。
6	F $	Feed	256個のNULLをパンチする。
7	G $	Gain	エディトバッファ領域の上限を指定する。
8	H $	Home	制御モニターに移す。
9	I strings $	Insert	stringsを現在のエディトバッファに挿入する。
10	K $ nK $	Kill	現在のバッファポインタからn行削除する。
11	L ($) (−)nL ($)	Line	バッファポインタを現在の行の先頭に移す。 バッファポインタを現在の位置よりn行移動させる。
12	M ($) (−)nM ($)	More	バッファポインタを現在の位置よりn文字移動させる。
13	N $	Next	エディトバッファの内容をすべてパンチし、次の紙テープ入力を行う。
14	O $	Out	FPVFのコマンドを行なう。
15	P $ nP $	Punch	エディトバッファの内容をすべてパンチする。 現在のエディトバッファからn行パンチする。
16	Q $	Quit	エディトバッファを初期化する。
17	R $	Rest	エディトバッファの未使用の領域を表示する。
18	S strings $	Search	stringsを現在のバッファポインタより捜していく。
19	T ($) nT ($)	Type	現在のエディトバッファよりn行タイプする。
20	U $	Unit	ASCIIコードのFF(0CH)をパンチする。
21	V $	Volume	ASCIIコードのEOT(04H)をパンチする。
22	W $	Write	エディトバッファの内容をすべてタイプする。
23	X $	Cross	アセンブラに制御を移す。
24	Y $	Yield	FPUVF$のコマンドを行なう。
25	Z ($) nZ ($)		バッファポインタを既入力の最後に移動させる。 バッファポインタを最後よりn行目に移動させる。

Z-80
アセンブラマニュアル

10

目　　次

ＳＭ−Ｂ−８０Ｄ　アセンブラ　ＡＳＭＢ

1. 序

ＡＳＭＢは、ワンボードのマイクロコンピュータ開発システムＳＭ−Ｂ−８０Ｄのもとで動作する紙テープベースのＺ８０レジデントアセンブラである。

2. ローディング

ＡＳＭＢはＥＤＩＴとともに一本の紙テープで供給される。この紙テープのローディングにはＳＭ−Ｂ−８０Ｄのモニタコマンド'Ｌ'を用いて行なわれる。

3. メモリマップ

ＡＳＭＢとＥＤＩＴは１００Ｈ〜１９００Ｈにロードされる。

メモリの１９００Ｈ〜３ＦＦＦＨはＡＳＭＢはシンボルテーブルとしてＥＤＩＴはエディットバッファとして用いる。

4. 起　　動

ＡＳＭＢとＥＤＩＴの起動にＳＭ−Ｂ−８０Ｄのモニタコマンド'Ｇ'を用いて行なわれる。

```
SM-B-80D  V1.1
*A;G
ASMB  V1.1
OPTION:
```

SM-B-80D　V1.1

*E；G

EDIT　V1.1

*

5.　ハードウェアの構成

ASMBとEDITは次のハードウェアを必要とする。

- コンソールタイプライタ
- RAM　16Kバイト以上
- SM-B-80　Dボード
- SM-B-80　Dモニタ

6.　各モジュール

6.1　ソースモジュール

ソースモジュールとは、FFにより分割され最後にEOTで終るユーザーのソースプログラムである。

6.2　オブジェクトモジュール

オブジェクトモジュールとは、アセンブラによって出力されるインテル標準ヘキサデシマルフォーマットの紙テープである。

7.　アセンブリ言語

7.1　ステートメント

ステートメントとはCRで終る72カラム以内のアスキー文字で、73カラム以上は無視される。

また、LF、FF、EOT、CR、DELを除くコントロールキャラクタは許されない。

7.2　分　離　子

ステートメントの各要素、ラベル、オペコード、オペランドおよびコメントは分離子によって分割される。

ラベルとオペコードは'：'、スペース、HTのいずれかによってオペコードとオペランドはスペースまたはHTによってオペコードとコメント、オペランドとコメントは'；'に

よって分割される。

7.3 ラ ベ ル

ラベルは1文字以上の英数字より構成される。ただし、7文字以上の場合は最初の6文字を有効とみなす。また最初の文字は数字であってはならない。ラベルは1カラムより始めるか':'で区切らなければならない。':'で区切ることにより一つのステートメントにいくつものラベルをつけることができる。

例

```
LAB
ABC:
 DEF:
L123
A: B: C:
```

7.4 オペコード

オペコードは、プログラミングマニュアル参照。

7.5 擬似命令

○ ORG nn origin

ロケーションカウンタをnnにセットする。

```
ORG 1000H
```

○ ラベル: EQU nn equate

ラベルの値をnnにセットする。

```
ABC: EQU 1234H
```

○ DEFM taaa……t 〈b、b……〉 define message

ASCIIの文字列aaa……を定義する。

tは';'、CR以外のキャラクターで文字列を示す分離子である。また'〈'、'〉'で囲むことによりバイト定数を定義することができる。

```
DEFM 'HELLO'〈ODH、OAH〉
DEFM 〈CR、LF〉/SM-B-80D/
```

○ DEFB n, n…… define byte

バイト定数nを定義する。

```
DEFB 1, 0AH, 10011B
DEFB 'A', 37Q
```

○　DEFW　　nn, nn, ………　　define word

ワード定数nnを定義する。

DEFW　ABC, 1234H, LABEL－3

○　DEFS　　nn　　define storage

nnバイトのエリアを定義する。

DEFS　　10H

DEFS　　NBYTE

7.6　アセンブリ命令

○　TITLE　　s　　title

各ページの先頭にタイトルをリスティングする。

sはスペース以外の文字で始まる16文字以内の文字列

TITLE　TEST　PROGRAM

TEST　PROGRAM　　Z80　ASSEMBLER　V1.1　PAGE　1

○　LIST　　listing on

リスティングを開始する。

○　NLIST　　listing off

リスティングを中止する。

○　PAGE　　page

ページをかえる。

○　END　　end

ソースプログラムの終了を示す。

7.7　オペランド

○　レジスタ

A、B、C、D、E、H、L、AF、BC、DE、HL、SP、IX、IY、AF'

ただし表現式内の上記のシンボルはラベルとみなされる。

LD　　A, B　　Bレジスタの内容をAレジスタへロードする。

LD　　A, B＋0　　ラベルBの値をAレジスタへロードする。

○　レジスタ間接

(HL)、(BC)、(DE)、(SP)、(IX+d)、(IY+d)、(IX)、

(IY)、(C)

○　イミディエイト定数

nn

ＬＤ　ＨＬ，１２３４Ｈ　　ＨＬレジスタへ１６進数１２３４をロードする。

○　エクステンド

(nn)

ＬＤ　Ａ，（１２３４Ｈ）　　Ａレジスタへメモリの１２３４Ｈ番地の内容をロードする。

nn およびｄは次に述べる表現式で表わされる１６ビット定数である。ただしｄの値は－１２８より＋１２７までしかとりえず、１バイトイミディエイトオペランドの場合は下位バイトのみ有効とする。

7.7.1　表　現　式

表現式は次に示す項と演算子より構成され演算は左より順に行なわれる。

○　項

＄　　現在のロケーションカウンタ

ラベル　　7.3ラベル参照

定　数

１０進数　　最後に'Ｄ'をつける。無指定の場合も１０進数とみなされる。

１２３　　１２３Ｄ

２進数　　０と１で表わし最後に'Ｂ'をつける。

１０１１００Ｂ　　１０００Ｂ

８進数　　最後に'Ｏ'または'Ｑ'をつける。

１０７Ｏ　　１２３Ｑ

１６進数　　最後に'Ｈ'をつける。ただし最初の数が'Ａ'～'Ｆ'の場合、その前に'０'をつけなければならない。

文　字

'Ａ'（＝４１Ｈ）

○　演算子

演算子には次の６種がある。演算は符号なしの１６ビットで行なわれ、演算時でのオーバーフロー　アンダーフローは無視される。ただし零割りはエラーとなる。

＋	加算
－	減算
＊	乗算
／	除算
＆	論理積
/	論理和

－1	（0FFFFH）
12＋39	（51　）
0110B＆10010B	（00010B）
0110B/10010B	（11110B）
6＊2	（12　）
31／3	（10　）
3／6	（0　）
1＋2＊2	（9　）

7.8　コメント

文字定数以外の'；'以降はコメントとみなされアセンブルの対象外となる。

ABC：　LD　A，B　；LOAD ACC

；　COMNTENT　LINE

7.9　オブジェクト

オブジェクトのフォーマットはインテル標準ヘキサデシマルフォーマットに準ずる。SM－B－80Dモニタ参照。

7.10　アセンブルリスト

アセンブルリストはSM－B－80DのSOチャンネルより出力される。フォーマットを次に示す。

EESSSSS△AAAA△OOOOOOOO△　statment

EE	エラーメッセージ
SSSSS	ステートメント番号
AAAA	アドレス　16進
OOOOOOOO	オブジェクトコード　16進

323　E389　0610　LD　B，10H

8. 操作方法

ＡＳＭＢはＳＭ－Ｂ－８０Ｄモニタのソースインプットチャンネル（ＳＩ）よりソースプログラムを入力し、リスティング指定があればソースアウトプットチャンネルへ出力し、オブジェクト指定があればオブジェクトアウトプットチャンネル（ＯＯ）へ出力する。したがってＡＳＭＢ起動時にはこれらのチャンネルに適当なディバイスを割り当てなければならない。

ＳＭ－Ｂ－８０Ｄモニタ起動時には各チャンネルは次のように割り当てられている。

○ ＳＩ　　○ ＴＲ

○ ＳＯ　　○ ＴＴ

○ ＯＯ　　○ ＴＴ

ＡＳＭＢは、起動されると以下のメッセージをタイプアウトしてオプションの入力を待つ。

```
*A;G
ASMB V1.1
OPTION:
```

8.1 オプション

○ Ｄオプション　　date

アセンブルリストの日付を設定する。

```
OPTION:D
DATE:27-APL-77↙
```

Ｄオプションを指定すると１０文字以内の文字列をアセンブルリストの右隅に出力することができる。なお、文字列は何らチェックを受けないのでプログラマ名を入れることもできる。１１文字以上は無視される。

```
Z80 ASSEMBLER V1.1 PAGE 1 27-APL-77
```

○ Ｇオプション　　gain

シンボルテーブルの上限の確認、および変更を行なう。

```
OPTION:G
CORE LIMIT:3FFF△ 新しい上限（16進）
             |
          現在の上限
```

１６Ｋバージョンでは通常上限は３ＦＦＦとなっている。また、この値はモニタよりＡＳＭＢを起動するたびに３ＦＦＦにリセットされる。

○ Hオプション　　　　　home

モニタへもどる。

　　OPTION:H↲

　　SM-B-80D V1.1

　　*

○ Lオプション　　　　　listing

ソースアウトプットチャンネルへアセンブルリストを出力する。

○ Oオプション　　　　　object

オブジェクトアウトプットチャンネルへオブジェクトを出力する。オブジェクトのフォーマットはインテルヘキサデシマルフォーマットに準ずる。

○ Qオプション　　　　　quit

ASMBの初期化を行なう。

　　OPTION:Q↲

　　ASMB　V1.1

　　OPTION:

インコアモード(8.4参照)のリセットに用いる。

シンボルテーブルの上限は変化しない。

○ Sオプション　　　　　symbol　table

パス2の最後にシンボルテーブルをソースアウトプットチャンネルへ出力する。もしLオプションが指定されていればアセンブルリストの後に出力される。

　　Z80　ASSEMBLER　V1.1 PAGE　1

SYMBOL　TABLE

ABC　　0000　　DAGH　1234　………

○ Xオプション　　　　cross

エディタに制御をうつす。

　　OPTION:X↲

　　EDIT　V1.1

　　*

○ 1オプション　　　　one　pass　mode

ワンパスモードでアセンブルを行なう。

8.3参照

○　2オプション　　pass two

パス2のみ行なう。

8.2参照

8.2　通常のオペレーション

次にオペレーションの例を示す。

```
*A ; G
ASMB V1.1
OPTION: L S↲
PASS1 READY? ①
```

⇦ 紙テープをセットし、任意のキーを打つ。もしソースプログラム中にFFがあればアセンブラはリード動作をやめる。この時、コンソールには何も出力されない。
次のモジュールをセットして任意のキーを打つことによりリード動作は再開される。

アセンブルリストを出力する。②

⇦ もう一度最初のモジュールより上記の操作を行なう。

シンボルテーブルを出力する。

```
ERRORS DETECTED:  0           ⇦ エラーの総数
FREE CORE: 16000 BYTES        ⇦ 未使用のメモリ容量
OPTION: O 2↲                  ⇦ パス2のみ行なう。
PASS2 READY?          ③
```

⇦ ②の操作を行なう。

オブジェクトを出力する。

```
ERRORS DETECTED: 0
FREE CORE: 16000 BYTES
OPTION:
```

※　オプションを何も指定しなかった場合、エラーのあるステートメントだけコンソールアウトプットチャンネルに出力される。

8.3　ワンパスモード

通常アセンブルは、２パスで行なわれるが、すでに出現したラベルのみを参照するソースプログラムに対しては１パスで行なうことができる。

OPTION：　1　L↲

PASS1　READY?

⇦　①

アセンブルリストを出力する。

ERRORS　DETEOTED：　0

FREE　CORE：　16000　BYTES

8.4　インコアモード

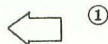
SM-B-80Dのアセンブラとエディタは一つのモジュールで構成されているため、エディタで作成または修正したソースプログラムを紙テープに出さずにメモリ内でアセンブルすることができる。また、アセンブルエラーがあればエディタにもどって（×オプション）、修正することもできる。

これによりまったく紙テープを介さずにアセンブルエラーを修正したソースプログラムおよびオブジェクトプログラムを作成することができる。

*E;G

EDIT　V1.1

*I

*X$$

ASMB　V1.1

OPTION：↲

PASS1　COMPLETED

ERRORS　DETECTED：　2

FREE　CORE：　932　BYTES

OPTION：X↲

EDIT　V1.1

*

9. エラーメッセージ表

B　bad character error

ソースプログラム内で許されていないキャラクタを入力した。

その文字は無視される。

D　dupulicate label refference error

多種定義されたラベルを参照した。

E　unfound END statement

ＥＮＤ文がない。

F　format error

フォーマットに誤りがある。

L　label error

ラベル内に英数字以外の文字が含まれていた。

最初の文字が英字でなかった。

M　multiple deffinition error

同一ラベルが複数回定義された。

O　opecode error

誤ったオペコードを入力した。

P　phase error

パス１とパス２でラベルの値が異なる。

Q　questionable error

オペランドの種類または組合せに誤りがある。

R　range error

許される範囲を超える値を入力した。

例えば'ＪＲ'でのｅの値が－１２６～＋１２９を超えた場合。

S　syntax error

構文に誤りがある。

T　symbal table overflow

シンボルテーブルがオーバーフローした。処理は続行される。

U　undefined label error

未定義のラベルを参照した。

SM-4 クロスアセンブラマニュアル

11

目　　　　　次

ＳＭ－４　クロスアセンブラ　ＸＳＭ４

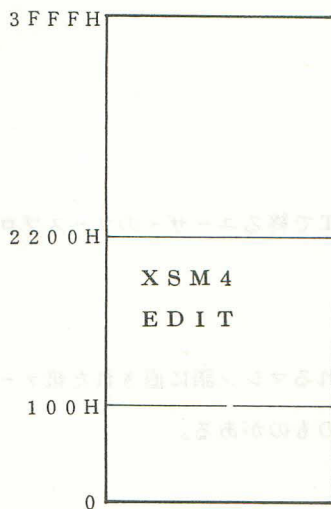
1. はじめに

クロスアセンブラＸＳＭ４は、ワンボードマイクロコンピュータ開発システムSM-B-80Dのもとで動作する紙テープベースのワンチップマイクロコンピュータＳＭ－４用のクロスアセンブラである。

2. ローディング

ＸＳＭ４は、ＸＳＭ４用のＥＤＩＴと共に一本の紙テープで供給される。この紙テープのローディングには、SM-B－８０Ｄのモニタコマンド'Ｌ'を用いて、次のようにキー入力する。

(＊) ；Ｌ
キー入力

3. メモリマップ

ＸＳＭ４とＥＤＩＴは、１００～２２００Ｈにロードされる。

メモリの２２００Ｈ～３ＦＦＦＨはＸＳＭ４のシンボルテーブルとして、またＥＤＩＴのエディットバッファとして用いる。

4. ハードウェアの構成

ＸＳＭ４とＥＤＩＴは次のハードウェアを必要とする。

1）　コンソールタイプライタ

2）　ＲＡＭ　　１６Ｋ以上

3）　SM-B－８０Ｄ　ボード

4）　SM-B－８０Ｄ　モニタ

5. 起　　動

ＸＳＭ４とＥＤＩＴの起動は、SM－B－８０Ｄのモニタコマンドの〝Ｇ〟を用いて行なわれる。

```
SM－B－80D　 V1．1
(*)A；G　（PROMバージョンの場合　．AS；G）
   キー入力

XSM4　V1．1
OPTION：
SMB－80D　V1．1
(*)E；G　（PROMバージョンの場合　．ED；G）
   キー入力

EDIT　V1．1
＊
```

6. 各モジュール

1）ソースモジュール

ソースモジュールとは、ＦＦにより分割され最後はＥＯＴで終るユーザーのソースプログラムである。

2）オブジェクトモジュール

オブジェクトモジュールとは、ＸＳＭ４によって出力されるマシン語に直された紙テープであり、エミュレーター用のものと、ＰＲＯＭライター用のものがある。

7. アセンブリ言語

1）ステートメント

ステートメントとは、ＣＲで終る７２カラム以内のアスキー文字で７３カラム以上は無視される。また、ＬＦ、ＦＦ、ＥＯＴ、ＣＲ、ＤＥＬを除くコントロールキャラクタは許されない。

2）分　離　子

ステートメントの各要素ラベル、オペコード、オペランド、及びコメントは分離子によっ

て分離される。

ラベルとオペコードは、ꞌ：ꞌ、スペース、HT※のいずれかによって、オペコードとオペランドは、スペースまたはHTによって、コメントとはꞌ；ꞌ、ꞌ！ꞌによって分離される。

3）ラ　ベ　ル

ラベルは1文字以上の英数字より構成される。又、文字は6文字までとし、7文字以上の場合は最初の6文字を有効とみなす。

ラベルは1カラムより始まらなければならない、ラベルの終りはꞌ：ꞌ、スペース、HTのどれか或はその組合せによって区切られる。

※HTはHorizontal Tab（水平タブ）のこと。

〈例〉

```
LAB：   ADD
LAC     ADD
LAD  ：  ADD
123     ADD
```

4）オペコード

別紙SM－4 Instruction Set 参照

5）擬似命令

```
ORG   mn
 ORGP  mn
```

ロケーションを設定する命令である。書式は、行頭にスペースあるいはHTを入れる。

〈例〉

```
␣ORG␣××××
     ↑  ↑
     |  └─────── ステップ指定（16進数）
     └─────────────── ページ指定

␣ORGP␣××××
      ↑  ↑
      |  └─────── ステップ指定（ポリノミアルコード）
      └─────── ページ指定
```

なお、␣はスペース又はHT、或はその組合せでもよい。

ORG、ORGPによるロケーション指定は、任意に行なえる。

ORGとORGPの相違はステップを指定する下位2桁にあり、ORGは、通常の16進数で指定し、ORGPはポリノミアルコードに従った16進数で指定することを意味する。

ORG、ORGPのロケーション指定は4桁以内で行なわれ、上位桁の0は省略できる。

〈例〉

0Aページの2ステップ目（ロケーション0A30）を指定する。

␣ＯＲＧ␣０Ａ０２

␣ＯＲＧ␣Ａ０２

␣ＯＲＧＰ␣０Ａ３０

␣ＯＲＧＰ␣Ａ３０

ＳＭ－４のＲＯＭページは３５ページであるが、ＯＲＧまたは、ＯＲＧＰでの指定は図１の通りである。

ＥＱＵ（＝Ｄ）

ＥＱＸ（＝）

ラベルの値を定義する命令であり、ＥＱＵと＝Ｄは同一意味を持っており、ＥＱＸと＝は同一意味をもっている。

書式は、ラベルを行頭から書き次の様にかく。

〈例〉

```
ABC:  EQX  13
CDE   EQU  13
AK:   =    1F
AC    =D   24
```

ＥＱＵで示されるのは、１０進数であり、上例ではＣＤＥが１０進数の１３即ち、１６進数のＤをあらわすことになる。

又、ＥＱＸは１６進数での値を示している。

この命令で与えられる数は、１６進数のＦＦまでである。又、ＥＱＵ命令の場合、使用出来る文字は、数字に限られ、ラベルの使用および加減算は許されない。ＥＱＸ命令の場合には、ラベルおよび加減算の使用は許される。

ＤＢ（＃）

バイト定数を定義する。（１６進数で定義する。）

〈例〉

```
DB   34
#    2F
#54
```

３４、２Ｆ、５４を１６進数でそのままマシンコードなる。この上限はＦＦである。

6）アセンブリ命令

TITLE

各ページの先頭にタイトルをリスティングする。

書式は次の様である。

〈例〉

TITLE TEST PROGRAM

TEST PROGRAM SM−4 ASSEMBLER.....

タイトルに使用できる文字は、スペース以外の文字で始まる16文字以内の文字列である。

LIST

リスティングを開始する。

NLIST

リスティングを禁止する。

PAGE

ページをかえる。

END

ソースプログラムの終了を示す。

7）オペランド

オペランドは少なくとも1個のスペースあるいはHTを前置して書かねばならない。

オペランドの先頭文字が0〜9、A〜Fの場合は、その数字、文字列は16進数とみなされる。従って、0〜9、A〜Fで始まるラベルの場合は必ず"（ダブルクォート）を付す必要がある。

〈例〉

```
AA   ⋮  EQX  4＋"ABC＋1     (＝9)
ABC     EQX  3 ＋ 1         (＝4)
     ⋮
EA      EXC  "ABC
        DTA  3
        TR0  "LBC
          ⋮
LBC:    LB   A
        TR0  "EA＋1
CDE:    TR0  "EA＋"ABC−2
```

オペランド内に書かれる記号は、下の様である。

"（ダブルクォート）　後続する文字、数字列をラベルとみなす。

＋（プラス）　後続する文字、数字列を前置する値に加えて算出する。

－（マイナス）　後続する文字、数字列を前置する値から減じて算出する。

．（ポイント）　現在のアドレスをオペランドとする。

8）コメント

；（セミコロン）、！（感嘆符）以降はコメントとみなされる。行頭が"＊"の場合、リスト作成時には24コラム目（ソースの先頭より7コラム目）におかれる。

9）マクロジャンプ命令

マクロジャンプ命令は、ラベルを使用してフィールド間、ページ間をジャンプする場合に便利な命令である。

マクロジャンプ命令には次のものがある。

␣．JPS␣"ABC　アドレスABCへのジャンプ命令を2バイトで作成し出力する。

␣．JPL␣"DEF　アドレスDEFへのジャンプ命令を3バイトで作成し出力する。

␣．JSS␣"GHI　アドレスGHIへのサブルーチンジャンプ命令を2バイトで作成し出力する。

␣．JSL␣"JKR　アドレスJKRへのサブルーチンジャンプ命令を3バイトで作成し出力する。

ジャンプのために、バイト数が不足あるいは、過剰である場合はFエラーとなる。

このマクロジャンプ命令は、サブルーチンページ（18xy～1Fxy、SOxy～S3xy）では使用できない。

8. アセンブルリスト

アセンブルリストはSM-B-80 DのSOチャンネルより出力される。

フォーマットは次の様である。

EESSSSS␣AAAA␣OOC␣Statment

EE　エラーメッセージ

SSSSS　ステートメント番号

AAAA　アドレス

ＯＯ　　　オブジェクトコード

Ｃ　　　コーテーションまたはスペース

なお２バイト、３バイト命令の場合は２バイト目、３バイト目と順々に行をかえて表記される。

〈例〉

Ｆ　323　0120　08　ＡＢ：　ＡＤＤ　ＢＣ ←―Ｆエラー

9. 操　作

ＸＳＭ４は、SM-B－８０Ｄモニタ（ＬＨ８Ｓ０３）のソースインプットチャンネル(ＳＩ)より、ソースプログラムを入力し、リスティング指定およびオブジェクト指定に対してソースアウトプットチャンネル（ＳＯ）、オブジェクトアウトプットチャンネル（ＯＯ）へ出力する。したがって、ＸＳＭ４起動時には、これらのチャンネルに適当なデバイスを割り当てる必要がある。

ＳＭＢ－８０Ｄモニタ起動時には、各チャンネルは次のように割りあてられる。

．ＳＩ　　．ＴＲ　（テレタイプテープリーダ）

．ＳＯ　　．ＴＴ　（テレタイプタイプヘッド）

．ＯＯ　　．ＴＴ　（　　　〃　　　）

ＸＳＭ４は、起動されると以下のメッセージをアウトプットしてオプションの入力を待つ。

＊Ａ；Ｇ　（ＰＲＯＭバージョンの場合　．ＡＳ；Ｇ）

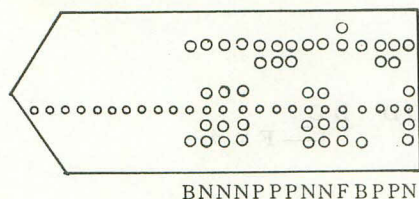
ＳＭ－４　Ｖ１．１

ＯＰＴＩＯＮ：

1）　オプション

Ｂ　　オプション　　ＢＮＰＦ

マシコードをＢＮＰＦフォーマットで出力する。出力順序は、ポリノミアルコードの順に従う。７ステップ毎におよび１ページ毎にＣＲＬＦを挿入して見やすくしている。また、各ページのステップは６３であるが、１ページの最後に００を出力し、各ページ６４バイトとして出力を行なう。

0 0 0 1 1 1 0 0　　1 1 0 0 1 1 0 0

D　オプション　　Date

アセンブルリストの日付を設定する。

ＯＰＴＩＯＮ：Ｄ

ＤＡＴＥ：　２１－ＪＵＮ－７８

ＯＰＴＩＯＮ：

Dオプションを指定すると、ただちに‘ＤＡＴＥ：’と印字を行なう。これに対して、１０文字以内の文字列を入力することができる。１１文字以上の文字は無視される。入力は、ＣＲで終了し、オプションの指定を続ける。入力された文字列は、アセンブラリストの右上隅に出力される。

ＳＭ－４　ＡＳＳＥＢＬＥＡＲ　Ｖ１．１　ＰＡＧＥ　１　２１－ＪＵＮ－７８

G　オプション　　Gain

シンボルテーブルの上限の確認、および変更を行なう。

ＯＰＴＩＯＮ：　Ｇ

ＣＯＲＥ　ＬＩＭＩＴ：　３ＦＦＦ␣××××↲

現在の上限　　新しい上限（１６進数でキー入力）
（変更なしの時　（ＣＲ）のみでよい）

上限の値はモニタよりＸＳＭ４を起動するたびに３ＦＦＦにセットされる。

なお、Ｄオプション、Ｇオプションでキー入力を取り消したい場合には、'］'キーを入力する。

H　オプション　Home

モニタへ制御をうつす。

ＯＰＴＩＯＮ：Ｈ↲

ＳＭ－Ｂ－８０Ｄ　Ｖ１．１
＊

L　オプション　Listing

ソースアウトプットチャンネル（．ＳＯ）へアセンブルリストを出力する。

O　オプション　Object

オブジェクトアウトプットチャンネル（．ＯＯ）へオブジェクトを出力する。オブジェクトのフォーマットは、シャープＳＭシリーズオリジナルフォーマットである。

★シャープＳＭシリーズオリジナルフォーマット

２バイトの１６進マシンコードと、１バイトのＣＲコードとの１組が１ステップ分として出力される。各コードはＡＳＣＩＩコードで出力され、ビット７には、許される機械仕様の場合すべて１がたてられる。

オブジェクトファイルの終端部には、＊（アスタリスク）が付加される。下図に出力される紙テープのフォーマットの概略図を示す。オブジェクトテープは、００００番地からＳ３３Ｅ番地までの全ステップ分が連続して出力される。オブジェクトの出力順序は、実行順（書き下し順）であり、命令のかかれなかった番地には、００が出力される。

P　オプション　PROM Format

オブジェクトアウトプットチャンネル（．ＯＯ）へオブジェクトを出力する。オブジェクトのフォーマットは、ブライトロニクス（Brightronics）のバイナリーフォーマットで、出力順序はポリノミアルコードの順に従って出力される。各ページのステップは６３ステップであり、６４ステップに００を出力し、各ページ６４ステップとして出力する。

★ブライトロニクスのバイナリーフォーマット

２バイトの１６進マシンコードと、１バイトのスペース（２０Ｈ）コードとが１ステップ分として出力される。各コードはＡＳＣＩＩコードで出力される。

Q　オプション　Quit

ＸＳＭ４の初期化を行なう。インコアモードの時のリセットにも用いられる。これによって、Ｇオプションの指定によるシンボルテーブルの上限には、何ら変化をうけない。

ＯＰＴＩＯＮ：　Ｑ↲

ＳＭ－４　Ｖ１．１

ＯＰＴＩＯＮ：

S　オプション　Symbol Table

シンボルテーブルを、ソースアウトプットへ出力する。もしＬオプションと共に指定されれば、アセンブルリストの出力の後に出力される。

シンボルテーブルリストには、シンボルのもつ値がアドレスの時には、４ケタで出力さ

れ、EQU、EQXによって定義された値は、2ケタで出力される。

SM－4　ASSEMBLER　V1.1　PAGE　1

SYMBOL　TABLE

ABC　0000　ABE　S123　AK　␣␣3F

アドレス　アドレス　数値

X　オプション　Cross

エディタに制御をうつす

OPTION：　X↲

EDIT　V1.1
＊

1　オプション　1 pass mode

ワンパスモードでアセンブルを行なう。

2　オプション　pass two

パス2のみ行なう。

2）通常のオペレーション

次にオペレーションの例を示す。

＊A；G

SM－4　V1.1

OPTION：　L　S↲

PASSI　READY?　①

紙テープをセットし任意のキーを打つ。もし、ソースプログラムにFFがあれば、アセンブラはリード動作をやめる。

PASS1　COMPLETED

PASS2　READY?

この時コンソールには何も出力されない。次のモジュールをセットして任意のキーを打つことによりリード動作は、再開される。

②

もう1度、①の操作を行なう。

アセンブルリストを出力する

シンボルテーブルを出力する

ERRORS　DETECTED：　n　←　エラーの総数

FREE　CORE：　8700　BYTES　←　未使用のメモリ容量

OPTION：　O↲　←　オブジェクトを出力する

オブジェクトを出力する

OPTION：

※ オブジェクトのみの時（O、B、Pオプション）、あるいは指定なしの場合は、.SOチャンネルより、エラーのあるラインのみを出力する。

※ 一度アセンブリした後、オブジェクトは、次にアセンブルするまで保存される。又、オプション'Q'、'X'、'H'によっても破壊される。

従って、アセンブルした後、'O'、'B'、'P'のみ指定をしたならば、直ちにオブジェクトを出力する。出力の順は、'O'、'P'、'B'オプションの順となる。

3）ワンパスモード

通常アセンブルは、2パスで行なわれるがすでに出現したラベルのみを参照するソースプログラムに対しては、1パスで行なうことができる。

OPTION：　1　L↲

PASS1　READY?

⇦ ①の操作を行なう。

アセンブルリストを出力する。

ERRORS　DETECTED：　0

FREE　CORE：　8700　BYTES

4）インコアモード

SM-B-80 DのXSM4とEDITは、一つのモジュールで構成されているため、エディタで作成または修正したソースプログラムを紙テープに出力せずにメモリ内でアセンブルすることができる。また、アセンブルエラーがあればエディタにもどって（Xオプション）修正することもできる。

これにより、まったく紙テープを介さずにアセンブルエラーを修正したソースプログラムとオブジェクトプログラムを作成することができる。

＊E；G　（PROMバージョンの場合　.ED；G）

EDIT　V1.1

＊I………

＊X$$

SM-4　V1.1

OPTION：↲

PASS1　COMPLETED

ERRORS DETECTED: 2

FREE CORE: 846 BYTES

OPTION: X↲

EDIT V1.1

*

10. エラーメッセージ表

A address error

このアドレスは、すでに使用されている。

B bad character error

ソースプログラム内で許されないキャラクタを入力した。

その文字は無視される。

C constant error

数値に誤りがある。

D duplicate label reference error

多重定義されたラベルを参照した。

E unfound END statement

END文がない。

F format error

フォーマットに誤りがある。

L label error

ラベル内に英数字以外の文字が含まれていた。

最初の文字が、英数字でなかった。

M multiple definition error

同一ラベルが複数個定義された。

O opecode error

誤ったオペコードを入力した。

P phase error

パス1とパス2でラベルの値が異なる。

R range error

許される範囲を超える値を入力した。

S　syntax error

構文に誤りがある。

T　symbol table overflow

シンボルテーブルがオーバーフローした。

ラベルは定義されないが、処理は続行される。

U　undefined label error

未定義のラベルを参照した。

J　jump error

ジャンプ出きない所へジャンプしようとした。

11. そ　の　他

1）コーテーションマーク（'）

アセンブルリストのマシンコードの後にフラグ"'"（コーテーションマーク）がたてられることがあり、これは次の事を意味する。

'が付されるインストラクションは、TR0、TR1の2つである。

(1)　TR0、TR1命令で'が付されているのは、ページ外ジャンプである事を示している。但し、TR1命令で10ページへジャンプする場合は、'は付されない。又、TR1命令でロケーションがサブルーチンページにある場合も、'は付されない。

2）C　エラー

マクロジャンプ命令、TR0、TR1、JMP命令以外でアドレスとして定義されたラベルを用いた場合、Cエラーとなる。この場合、用いることができるのはEQU、EQXで定義したラベルのみである。

Cx フィールド ページ Pu CA	0		1
	0	1	＊
0000	00xy	10xy (サブルーチンの表紙)	S0xy (20xy)
0001	01xy	11xy	S1xy (21xy)
0010	02xy	12xy	S2xy (22xy)
0011	03xy	13xy	S3xy (23xy)
0100	04xy	14xy	
0101	05xy	15xy	
0110	06xy	16xy	
0111	07xy	17xy	
1000	08xy	18xy	
1001	09xy	19xy	
1010	0Axy	1Axy	
1011	0Bxy	1Bxy	
1100	0Cxy	1Cxy	
1101	0Dxy	1Dxy	
1110	0Exy	1Exy	
1111	0Fxy	1Fxy	

SC：ページ0000～0011の＊列（S0xy～S3xy）
SA：ページ1000～1011
SB：ページ1100～1111

註1） xyは16進数のステップを示す。

2） ORG(P)の指定なき場合、0000より始まる。

3） SA、SB、SCはサブルーチンページを示す。

4） 10xyからJMP命令でジャンプ可能なページを斜線で示す。

図1　　アドレス配置図

SM-B-80T
アプリケーションマニュアル(1)

12

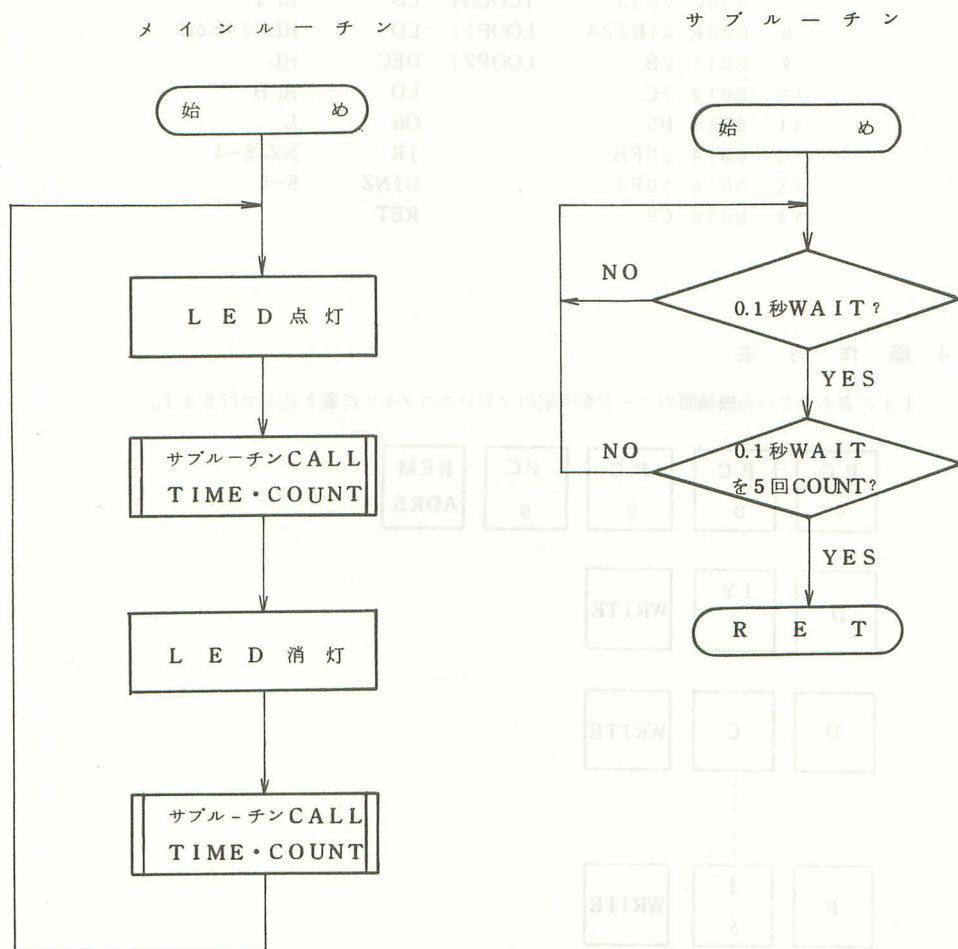
目　　　次

1.　リモート用ＬＥＤの点滅

1.1　概　　要

ＳＭ－Ｂ－８０Ｔに取り付けられているリモート端子用ＬＥＤを約１秒周期で点滅させせます。

このプログラムは、オーディオカセットのリモート端子に割り当てているＩ／Ｏポートを制御してＬＥＤを点滅させせます。

1.2　フローチャート

1.3 コーディング

```
   アドレス機械語                      アセンブリ言語
 1                                    ORG     0
 2  0000  D3DC           LOOP:        OUT     (0DCH), A
 3  0002  CD0C00                      CALL    TCOUNT
 4  0005  DBDC                        IN      A, (0DCH)
 5  0007  CD0C00                      CALL    TCOUNT
 6  000A  18F4                        JR      $-10
 7  000C  0605           TCOUNT:      LD      B, 5
 8  000E  21E624         LOOP1:       LD      HL, 24E6H
 9  0011  2B             LOOP2:       DEC     HL
10  0012  7C                          LD      A, H
11  0013  B5                          OR      L
12  0014  20FB                        JR      NZ, $-3
13  0016  10F6                        DJNZ    $-8
14  0018  C9                          RET
```

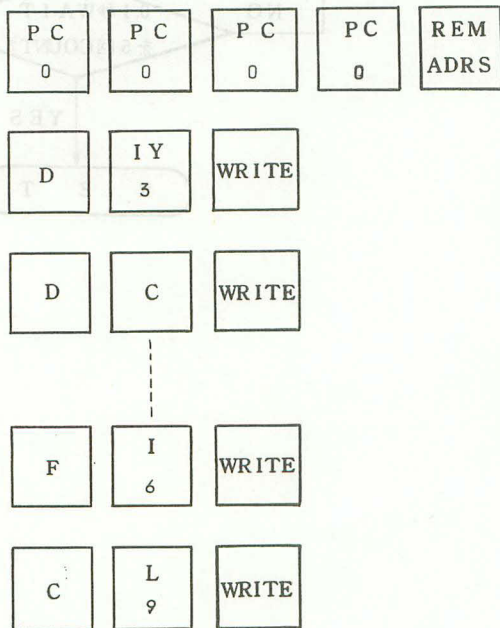
1.4 操　作　方　法

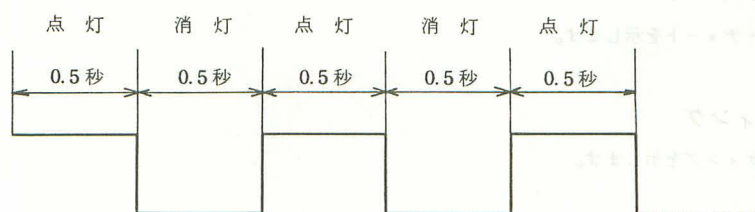
1.3 に書かれている機械語のコードを所定のアドレスのメモリに書き込んで行きます。

```
[PC 0] [PC 0] [PC 0] [PC 0] [REM ADRS]

[D] [IY 3] [WRITE]

[D] [C] [WRITE]
      |
[F] [I 6] [WRITE]

[C] [L 9] [WRITE]
```

プログラムの書き込みが終了しましたら、次のキー操作で実行させます。

プログラムを実行しますと次のタイミングでＬＥＤを点滅させます。

. デジタル・クロック

2.1 概　　要

CPU（LH－0080）の命令サイクルをカウントして、LED表示部に時刻を表示するクロックのプログラムです。

このプログラムは、CPUの命令サイクルをカウントし、1／100sec ごとにLEDの表示を変更していきます。また、クロックの一時停止機能、タイマ機能もあります。（基準は、システム・クロック2.4576MHz を利用）

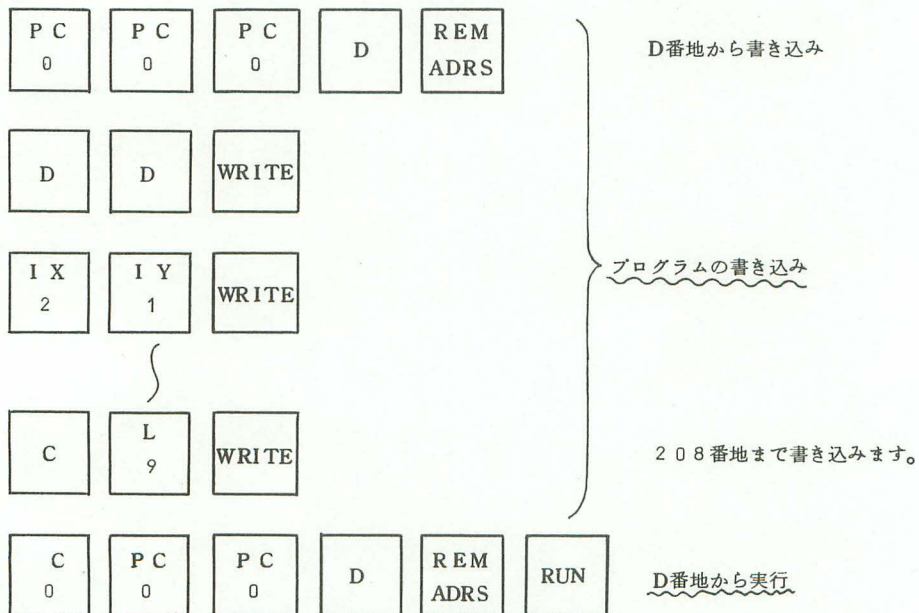
2.2 フローチャート

別図にフローチャートを示します。

2.3 コーディング

別表にコーディングを示します。

2.4 プログラムの書き込みと実行

2.4.1 プログラムを、キーから入力し、実行させます。

2.4.2　初　期　設　定

a．タイマのスイッチON，OFF設定

b．スタート時刻の設定、計時

RUN　　計時のスタート

2.4.3　各キーの役割の説明

キー	説明
RUN	計時のスタート。
STEP	計時のストップ。これよりストップウオッチとしても用いることができます。
LOAD INC	表示の0クリア。（タイマのON，OFFは変更されません。）
STOR DEC	リセット。（タイマについてもすべてリセットされます。）
SHIFT	タイマの設定のリセット。このキーに続いて“REM”，“REG”キーを押すことによりリセットされます。

キー	説明
REG' / REG	タイマのOFF時刻の表示、また、"WRITE"，"SHIFT"キーを押した後にこのキーを入力しますと、タイマのOFF時刻のセット、リセットが行えます。
REM / ADRS	タイマのON時刻の表示、また、"WRITE"，"SHIFT"キーを押した後にこのキーを入力しますと、タイマのON時刻のセット、リセットが行えます。
WRITE	タイマのON，OFF時刻のセット、このキーに続いて、"REG"，"REM"キーを入力すると、表示されている時刻が、タイマのON，OFF時刻としてセットされます。
A ～ F	使用しないキーです。
PC 0 ～ L 9	表示の右端から順に入力されて、左へ1つづつシフトしていきます。

注1. タイマのON，OFFがセットされていない場合、次の様な表示となります。

2. クロックの修正は、0E3Hの値を変更することにより可能です。+1で約5.3 μs 遅くなります。

フローチャート

始め
初期設定
1
表示
2
キー入力
RUN ?
YES
3
NO
SHIFT ?
YES
4
NO
REM ?
YES
5
NO
REG ?
YES
6
NO
WRITE ?
YES
7
NO
LOAD ?
YES
表示の "0" クリア
NO
STOR ?
YES
NO
"0" ~ "9" ?
NO
YES
左へシフト

3
GO
（時間の加算）
表　示
キー入力？
NO
YES
STEP？
YES
1
NO
タイマ
ON SETされているか？
NO
YES
設定時刻か？
NO
YES
タイマのスイッチ
ON
タイマ
OFF SETされているか？
NO
YES
設定時刻か？
NO
YES
タイマのスイッチ
OFF
時間の
調　整
4
キー入力
REM？
YES
タイマONの
リセット
NO
REG？
YES
タイマOFFの
リセット
NO
SHIFT？
YES
NO
1
7
キー入力
REM？
YES
タイマONの
セット
NO
REG？
YES
タイマOFFの
セット
NO
SHIFT？
YES
NO
1

5
タイマON
セット？
NO
YES
タイマONの
内容の表示
セットされていない
旨の表示
2

6
タイマOFF
セット ？
NO
YES
タイマOFFの
内容の表示
セットされていない
旨の表示
2

001 <<CLOCK>>

```
 1                                  TITLE    <<CLOCK>>
 2                          ;       ***** MEMORY ****
 3                                  ORG      0
 4 0000                     NUM     DEFS     1
 5 0001                     SEC     DEFS     1
 6 0002                     MIN     DEFS     1
 7 0003                     HOUR    DEFS     1
 8 0004                     REM0    DEFS     1
 9 0005                     REM1    DEFS     1
10 0006                     REM2    DEFS     1
11 0007                     REM3    DEFS     1
12 0008                     REM4    DEFS     1
13 0009                     REM5    DEFS     1
14 000A                     REM6    DEFS     1
15 000B                     REM7    DEFS     1
16 000C                     REM8    DEFS     1
17                          ;       ******* MAIN *******
18 000D DD210000            MAIN:   LD       IX,0
19 0011 FD21CDFF                    LD       IY,SEGBUF
20 0015 DD360400                    LD       (IX+REM0),0
21 0019 3199FF              START:  LD       SP,SPSET
22 001C 060D                        LD       B,13
23 001E 210000              CLER:   LD       HL,NUM
24 0021 AF                          XOR      A
25 0022 77                  HCLER:  LD       (HL),A
26 0023 23                          INC      HL
27 0024 10FC                        DJNZ     HCLER
28 0026 CDB401              TM1:    CALL     TIME
29 0029 CD3FE3              TM3:    CALL     KEYIN
30 002C FE13                        CP       CRUN
31 002E 2826                        JR       Z,RUN
32 0030 FE17                        CP       CSFT
33 0032 CAC400                      JP       Z,SFT
34 0035 FE16                        CP       CREM
35 0037 2863                        JR       Z,ON
36 0039 FE10                        CP       REG
37 003B 286D                        JR       Z,OFF
38 003D FE12                        CP       STOR
39 003F 28CC                        JR       Z,MAIN
40 0041 FE11                        CP       LOAD
41 0043 2004                        JR       NZ,TM2
42 0045 0604                        LD       B,4
43 0047 18D5                        JR       CLER
44 0049 FE15                TM2:    CP       WRITE
45 004B 2824                        JR       Z,WRT
46 004D FE0A                        CP       9+1
47 004F 30D5                        JR       NC,TM1
48 0051 CDEB00                      CALL     SHIFT0
49 0054 18D0                TM11:   JR       TM1
50                          ;       ******* RUN *******
```

```
51 0056 CDF500     RUN:    CALL    GO
52 0059 CDB401             CALL    TIME
53 005C CDD601             CALL    SCAN
54 005F CDD601             CALL    SCAN
55 0062 00                 NOP
56 0063 00                 NOP
57 0064 00                 NOP
58 0065 FE51               CP      51H
59 0067 28EB       TM12:   JR      Z,TM11
60 0069 CD5101     RUN2:   CALL    REMOTE
61 006C CDF200             CALL    COUNT
62 006F 18E5               JR      RUN
63                 ;       ****** WRITE ******
64 0071 CD3FE3     WRT:    CALL    KEYIN
65 0074 FE16               CP      CREM
66 0076 280A               JR      Z,WRON
67 0078 FE10               CP      REG
68 007A 2817               JR      Z,WROFF
69 007C FE15               CP      WRITE
70 007E 28F1               JR      Z,WRT
71 0080 18A7       TM13:   JR      TM3
72                 ;       **** WRITE ON OFF *****
73 0082 DDCB04C6   WRON:   SET     0,(IX+REMO)
74 0086 110500            LD      DE,REM1
75 0089 010400     WRON1:  LD      BC,4
76 008C 210000            LD      HL,NUM
77 008F EDB0               LDIR
78 0091 18ED               JR      TM13
79 0093 DDCB04CE   WROFF:  SET     1,(IX+REMO)
80 0097 110900            LD      DE,REM5
81 009A 18ED               JR      WRON1
82                 ;       ******* ON OFF *****
83 009C DDCB0446   ON:     BIT     0,(IX+REMO)
84 00A0 2813               JR      Z,ON2
85 00A2 210500            LD      HL,REM1
86 00A5 CDB701     ON1:    CALL    TIME1
87 00A8 18D6       TM14:   JR      TM13
88 00AA DDCB044E   OFF:    BIT     1,(IX+REMO)
89 00AE 2805               JR      Z,ON2
90 00B0 210900            LD      HL,REM5
91 00B3 18F0               JR      ON1
92 00B5 21C0FF     ON2:    LD      HL,SEGBUF
93 00B8 010408            LD      BC,804H
94 00BB 71         ON3:    LD      (HL),C
95 00BC 23                 INC     HL
96 00BD 10FC               DJNZ    ON3
97 00BF CDC201             CALL    TIME2
98 00C2 18E4       TM15:   JR      TM14
99                 ;       ******* SHIFT ON OFF *******
00 00C4 CD3FE3     SFT:    CALL    KEYIN
```

```
101 00C7 FE16              CP      CREM
102 00C9 280B              JR      Z,SFT1
103 00CB FE10              CP      REG
104 00CD 280D              JR      Z,SFT2
105 00CF FE17              CP      CSFT
106 00D1 28F1              JR      Z,SFT
107 00D3 C35400    TM16:   JP      TM11
108 00D6 DDCB0486  SFT1:   RES     0,(IX+REMO)
109 00DA 18F7              JR      TM16
110 00DC DDCB048E  SFT2:   RES     1,(IX+REMO)
111 00E0 18F1              JR      TM16
112                ;       ****** COUNT *******
113 00E2 06F5      COUNT:  LD      B,0F5H
114 00E4 10FE              DJNZ    $
115 00E6 DD210000          LD      IX,0
116 00EA C9                RET
117                ;       ***** SHIFT ******
118 00EB 210000    SHIFT0: LD      HL,NUM
119 00EE CD1EE3            CALL    SHIFT1
120 00F1 23                INC     HL
121 00F2 C31EE3            JP      SHIFT1
122                ;       ***** GO ******
123 00F5 210300    GO:     LD      HL,HOUR
124 00F8 7E                LD      A,(HL)
125 00F9 FE24              CP      24H
126 00FB 3018              JR      NC,NONGO
127 00FD 2B                DEC     HL
128 00FE 7E                LD      A,(HL)
129 00FF FE60              CP      60H
130 0101 3012              JR      NC,NONGO
131 0103 2B                DEC     HL
132 0104 7E                LD      A,(HL)
133 0105 FE60              CP      60H
134 0107 300C              JR      NC,NONGO
135 0109 2B                DEC     HL
136 010A 34        GO1:    INC     (HL)
137 010B 7E                LD      A,(HL)
138 010C B7                OR      A
139 010D 27                DAA
140 010E 77                LD      (HL),A
141 010F 2807              JR      Z,GO2
142 0111 060F      GONOP1: LD      B,15
143 0113 1825              JR      GONOP4
144 0115 C31900    NONGO:  JP      START
145 0118 3600      GO2:    LD      (HL),0
146 011A 23                INC     HL
147 011B 34                INC     (HL)
148 011C 7E                LD      A,(HL)
149 011D B7                OR      A
150 011E 27                DAA
```

```
151 011F 77                    LD      (HL),A
152 0120 FE60                  CP      60H
153 0122 2805                  JR      Z,GO3
154 0124 060A      GONOP2:     LD      B,10
155 0126 7E                    LD      A,(HL)
156 0127 1822                  JR      GONOP5
157 0129 3600      GO3:        LD      (HL),0
158 012B 23                    INC     HL
159 012C 34                    INC     (HL)
160 012D 7E                    LD      A,(HL)
161 012E B7                    OR      A
162 012F 27                    DAA
163 0130 77                    LD      (HL),A
164 0131 FE60                  CP      60H
165 0133 2808                  JR      Z,GO4
166 0135 0605      GONOP3:     LD      B,5
167 0137 00                    NOP
168 0138 1811                  JR      GONOP5
169 013A C34B01    GONOP4:     JP      GONOP5
170 013D 3600      GO4:        LD      (HL),0
171 013F 23                    INC     HL
172 0140 34                    INC     (HL)
173 0141 7E                    LD      A,(HL)
174 0142 B7                    OR      A
175 0143 27                    DAA
176 0144 77                    LD      (HL),A
177 0145 FE24                  CP      24H
178 0147 2805                  JR      Z,GO5
179 0149 0601                  LD      B,1
180 014B 10FE      GONOP5:     DJNZ    $
181 014D C9                    RET
182 014E 3600      GO5:        LD      (HL),0
183 0150 C9                    RET
184                ;           ***** REMOTE *****
185 0151 DD7E04    REMOTE:     LD      A,(IX+REMO)
186 0154 0F                    RRCA
187 0155 DC6B01                CALL    C,RMT01
188 0158 D4AD01                CALL    NC,RMT10
189 015B 0F                    RRCA
190 015C DD210400              LD      IX,4
191 0160 DC9401                CALL    C,RMT02
192 0163 D4AD01                CALL    NC,RMT10
193 0166 DD210000              LD      IX,0
194 016A C9                    RET
195 016B F5        RMT01:      PUSH    AF
196 016C CD7501                CALL    REMO0
197 016F 2039                  JR      NZ,RMT06
198 0171 D3DC                  OUT     (REM),A
199 0173 F1                    POP     AF
200 0174 C9                    RET
```

```
201 0175 3A0000   REM00:  LD      A,(NUM)
202 0178 DDBE05           CP      (IX+REM1)
203 017B 2021             JR      NZ,RMT03
204 017D 3A0100           LD      A,(SEC)
205 0180 DDBE06           CP      (IX+REM2)
206 0183 201D             JR      NZ,RMT04
207 0185 3A0200           LD      A,(MIN)
208 0188 DDBE07           CP      (IX+REM3)
209 018B 2019             JR      NZ,RMT05
210 018D 3A0300           LD      A,(HOUR)
211 0190 DDBE08           CP      (IX+REM4)
212 0193 C9               RET
213 0194 F5       RMT02:  PUSH    AF
214 0195 CD7501           CALL    REM00
215 0198 2010             JR      NZ,RMT06
216 019A DBDC             IN      A,(REM)
217 019C F1               POP     AF
218 019D C9               RET
219 019E 0607     RMT03:  LD      B,7
220 01A0 18A9             JR      GONOP5
221 01A2 0604     RMT04:  LD      B,4
222 01A4 18A5             JR      GONOP5
223 01A6 0601     RMT05:  LD      B,1
224 01A8 18A1             JR      GONOP5
225 01AA 23       RMT06:  INC     HL
226 01AB F1               POP     AF
227 01AC C9               RET
228 01AD C5       RMT10:  PUSH    BC
229 01AE 060E             LD      B,14
230 01B0 10FE             DJNZ    $
231 01B2 C1               POP     BC
232 01B3 C9               RET
233                       ;       ***** TIME *****
234 01B4 210000   TIME:   LD      HL,NUM
235 01B7 010400   TIME1:  LD      BC,4
236 01BA 11DFFF           LD      DE,DATA
237 01BD EDB0             LDIR
238 01BF CDF7E2           CALL    DISP
239 01C2 CDC901   TIME2:  CALL    DOT
240 01C5 CDD601           CALL    SCAN
241 01C8 C9               RET
242                       ;       ****** DOT *****
243 01C9 FDCB02FE DOT:    SET     7,(IY+T1)
244 01CD FDCB04FE         SET     7,(IY+T2)
245 01D1 FDCB06FE         SET     7,(IY+T3)
246 01D5 C9               RET
247                       ;       ****** SCAN ******
248 01D6 1600     SCAN:   LD      D,0
249 01D8 0E70             LD      C,70H
250 01DA 21D4FF           LD      HL,SEGBUF+7
```

```
251 01DD 3E80       SN1:   LD      A,80H
252 01DF D3D2              OUT     (PIOBD),A
253 01E1 7E                LD      A,(HL)
254 01E2 2B                DEC     HL
255 01E3 D3D0              OUT     (PIOAD),A
256 01E5 79                LD      A,C
257 01E6 D3D2              OUT     (PIOBD),A
258 01E8 DBD2              IN      A,(PIOBD)
259 01EA 2F                CPL
260 01EB E60F              AND     0FH
261 01ED 2804              JR      Z,SN2
262 01EF B1                OR      C
263 01F0 57                LD      D,A
264 01F1 1804              JR      SN3
265 01F3 2002       SN2:   JR      NZ,SN3
266 01F5 00                NOP
267 01F6 00                NOP
268 01F7 3E28       SN3:   LD      A,40
269 01F9 3D         SN4:   DEC     A
270 01FA 20FD              JR      NZ,SN4
271 01FC 79                LD      A,C
272 01FD D610              SUB     10H
273 01FF 4F                LD      C,A
274 0200 30DB              JR      NC,SN1
275 0202 3E80              LD      A,80H
276 0204 D3D2              OUT     (PIOBD),A
277 0206 7A                LD      A,D
278 0207 A7                AND     A
279 0208 C9                RET
280                 ;      ******* DEFINE ******
281      FFDF       DATA   EQU     0FFDFH
282      FFCD       SEGBUF EQU     0FFCDH
283      E33F       KEYIN  EQU     0E33FH
284      0010       REG    EQU     10H
285      0011       LOAD   EQU     11H
286      0012       STOR   EQU     12H
287      0013       CRUN   EQU     13H
288      0014       STEP   EQU     14H
289      0015       WRITE  EQU     15H
290      0016       CREM   EQU     16H
291      0017       CSFT   EQU     17H
292      E31B       SHIFT  EQU     0E31BH
293      E31E       SHIFT1 EQU     0E31EH
294      E324       SEGCON EQU     0E324H
295      E2F7       DISP   EQU     0E2F7H
296      FFE1       ADDR   EQU     0FFE1H
297      00D0       PIOAD  EQU     0D0H
298      00D2       PIOBD  EQU     0D2H
299      00DC       REM    EQU     0DCH
300      FF99       SPSET  EQU     0FF99H
```



```
301      0002       T1     EQU     2
302      0004       T2     EQU     4
303      0006       T3     EQU     6
304 0209                   END
```

<<CLOCK>> SYMBOL TABLE

```
ADDR   -FFE1   CLER   -001E   COUNT  -00E2   CREM   -0016
CRUN   -0013   CSFT   -0017   DATA   -FFDF   DISP   -E2F7
DOT    -01C9   GO     -00F5   GO1    -010A   GO2    -0118
GO3    -0129   GO4    -013D   GO5    -014E   GONOP1-0111
GONOP2-0124   GONOP3-0135   GONOP4-013A   GONOP5-014B
HCLER  -0022   HOUR   -0003   KEYIN  -E33F   LOAD   -0011
MAIN   -000D   MIN    -0002   NONGO  -0115   NUM    -0000
OFF    -00AA   ON     -009C   ON1    -00A5   ON2    -00B5
ON3    -00BB   PIOAD  -00D0   PIOBD  -00D2   REG    -0010
REM    -00DC   REM0   -0004   REM00  -0175   REM1   -0005
REM2   -0006   REM3   -0007   REM4   -0008   REM5   -0009
REM6   -000A   REM7   -000B   REM8   -000C   REMOTE-0151
RMT01  -016B   RMT02  -0194   RMT03  -019E   RMT04  -01A2
RMT05  -01A6   RMT06  -01AA   RMT10  -01AD   RUN    -0056
RUN2   -0069   SCAN   -01D6   SEC    -0001   SEGBUF-FFCD
SEGCON-E324   SFT    -00C4   SFT1   -00D6   SFT2   -00DC
SHIFT  -E31B   SHIFT0-00EB   SHIFT1-E31E   SN1    -01DD
SN2    -01F3   SN3    -01F7   SN4    -01F9   SPSET  -FF99
START  -0019   STEP   -0014   STOR   -0012   T1     -0002
T2     -0004   T3     -0006   TIME   -01B4   TIME1  -01B7
TIME2  -01C2   TM1    -0026   TM11   -0054   TM12   -0067
TM13   -0080   TM14   -00A8   TM15   -00C2   TM16   -00D3
TM2    -0049   TM3    -0029   WRITE  -0015   WROFF  -0093
WRON   -0082   WRON1  -0089   WRT    -0071
```

3.自　動　演　奏

3.1　概　　要

SM－B－80Tの応用として、音楽を連続して自動演奏するプログラム例を示します。

このプログラムは、演奏させたい曲の音符を2バイトのデータに変換して音符データエリアに格納しておき、この音符データに対応した周波数パルスをF／Fに接続したオーディオアンプに出力して演奏します。

3.2　フローチャート

別図にフローチャートを示します。

3.3　コーディング

別表にコーディングを示します。

3.4　音符データの作成

音符データは、次に示すようにメモリの2バイトを使用して下位1バイトに音階とオクターブを、上位1バイトに音符の長さをセットします。

このプログラムでは、音階として12平均律音階を、オクターブは3オクターブまで使用します。また、同じ曲を繰り返し演奏させるには、曲の終わりに一定の休符を入れた後、0をセットします。

上位アドレス　音符の長さ

オクターブの入力

0　(0000)　繰り返しのとき

1　(0001)　1オクターブ

2　(0010)　2オクターブ

4　(0100)　3オクターブ

8　(1000)　休符

＃（シャープ）の入力

0　そのままの音階

1　半音上げる

音階の入力

000　繰り返しのとき（このとき＃の入力も0にすること）

001　ド

010　レ

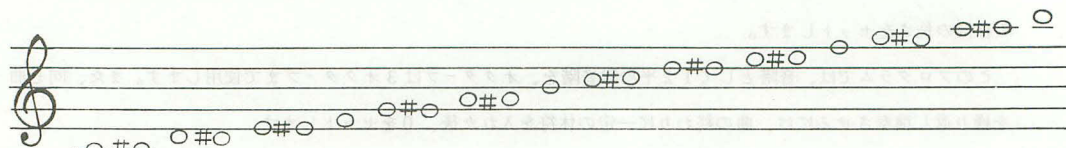
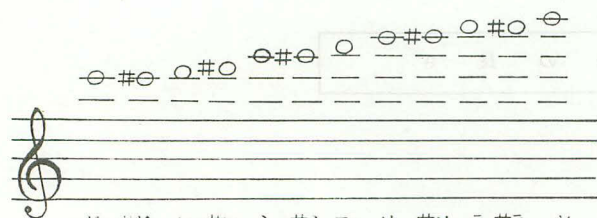
011　ミ

100　ファ

101　ソ

110　ラ

111　シ

五線符との対応

ハ長調の音階	ド	＃ド	レ	＃レ	ミ	＃ミ	ファ	ソ	＃ソ	ラ	＃ラ	シ	ド	＃ド	レ	＃レ	ミ	＃ミ	ファ	ソ	＃ソ	ラ	＃ラ	シ
絶対音階	ハ	変ハ	ニ	変ニ	ホ	変ホ	ヘ	ト	変ト	イ	変イ	ロ	ハ	変ハ	ニ	変ニ	ホ	変ホ	ヘ	ト	変ト	イ	変イ	ロ
データ	12	13	14	15	16	17	18	1A	1B	1C	1D	1E	22	23	24	25	26	27	28	2A	2B	2C	2D	2E

ド	＃ド	レ	＃レ	ミ	＃ミ	ファ	ソ	＃ソ	ラ	＃ラ	シ
ハ	変ハ	ニ	変ニ	ホ	変ホ	ヘ	ト	変ト	イ	変イ	ロ
42	43	44	45	46	47	48	4A	4B	4C	4D	4E

音符の長さ

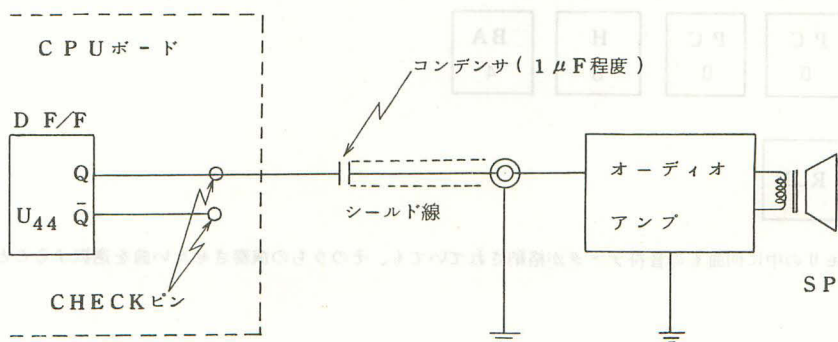
音符		データ（16進数）
	32分音符	1
	32分休符	1
	16分音符	2
	16分休符	2
	8分音符	4
	8分休符	4
	4分音符	8
	4分休符	8
	2分音符	10
	2分休符	10
	全音符	20
	全休符	20

（注）　データとして、32分音符を1として計算しています。他の符点音符などは、計算してください。

（例）　♩. ＝♩＋♪＝8＋4＝C（16進数）

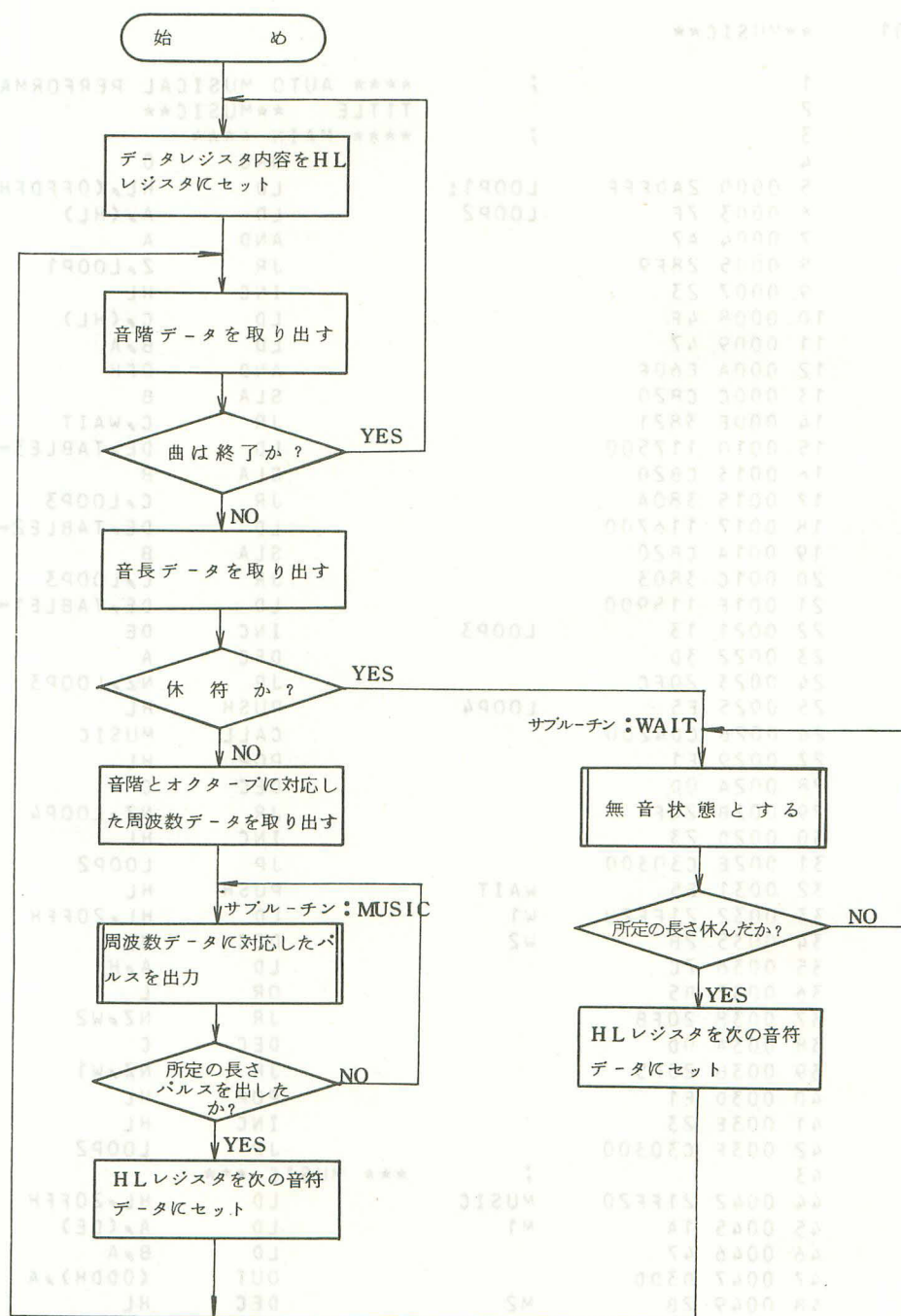
𝅗𝅥.. ＝𝅗𝅥＋♩＋♪＝10＋8＋4＝1C（16進数）

3.5 操作方法

このプログラムで使用するDタイプF／Fとオーディオアンプは次のように接続します。

次のキー操作でコーディングリスト上の機械語コードを所定のメモリに書き込みます。

プログラムの書き込みが終了しましたら次のキー操作で実行させます。ただし、自動演奏させたい曲の音符データはメモリの0084H番地から格納されているものとします。

メモリの中に何曲もの音符データが格納されていても、そのうちの演奏させたい曲を選択することができます。

自動演奏フローチャート

```
MUSIC.TS    ASMBL'D BY Z80 ASSEMBLER REV-A.1    06,20,'78
        TITLE    **MUSIC**

001     **MUSIC**

         1                          ;         **** AUTO MUSICAL PERFORMANCE ****
         2                                    TITLE    **MUSIC**
         3                          ;         **** MAIN ****
         4                                          ORG     0
         5  0000  2ADFFF            LOOP1:          LD      HL,(OFFDFH)
         6  0003  7E                LOOP2           LD      A,(HL)
         7  0004  A7                                AND     A
         8  0005  28F9                              JR      Z,LOOP1
         9  0007  23                                INC     HL
        10  0008  4F                                LD      C,(HL)
        11  0009  47                                LD      B,A
        12  000A  E60F                              AND     OFH
        13  000C  CB20                              SLA     B
        14  000E  3821                              JR      C,WAIT
        15  0010  117500                            LD      DE,TABLE3-2
        16  0013  CB20                              SLA     B
        17  0015  380A                              JR      C,LOOP3
        18  0017  116700                            LD      DE,TABLE2-2
        19  001A  CB20                              SLA     B
        20  001C  3803                              JR      C,LOOP3
        21  001E  115900                            LD      DE,TABLE1-2
        22  0021  13                LOOP3           INC     DE
        23  0022  3D                                DEC     A
        24  0023  20FC                              JR      NZ,LOOP3
        25  0025  E5                LOOP4           PUSH    HL
        26  0026  CD4200                            CALL    MUSIC
        27  0029  E1                                POP     HL
        28  002A  0D                                DEC     C
        29  002B  20F8                              JR      NZ,LOOP4
        30  002D  23                                INC     HL
        31  002E  C30300                            JP      LOOP2
        32  0031  E5                WAIT            PUSH    HL
        33  0032  21FF20            W1              LD      HL,20FFH
        34  0035  2B                W2              DEC     HL
        35  0036  7C                                LD      A,H
        36  0037  B5                                OR      L
        37  0038  20FB                              JR      NZ,W2
        38  003A  0D                                DEC     C
        39  003B  20F5                              JR      NZ,W1
        40  003D  E1                                POP     HL
        41  003E  23                                INC     HL
        42  003F  C30300                            JP      LOOP2
        43                          ;          *** MUSIC ***
        44  0042  21FF20            MUSIC           LD      HL,20FFH
        45  0045  1A                M1              LD      A,(DE)
        46  0046  47                                LD      B,A
        47  0047  D3DD                              OUT     (ODDH),A
        48  0049  2B                M2              DEC     HL
        49  004A  7C                                LD      A,H
        50  004B  B5                                OR      L
```

```
51 004C C8                                  RET     Z
52 004D 10FA                                DJNZ    M2
53 004F 1A                                  LD      A,(DE)
54 0050 47                                  LD      B,A
55 0051 DBDD                                IN      A,(0DDH)
56 0053 2B          M3                      DEC     HL
57 0054 7C                                  LD      A,H
58 0055 B5                                  OR      L
59 0056 C8                                  RET     Z
60 0057 10FA                                DJNZ    M3
61 0059 18EA                                JR      M1
62                  ;            **** TABLE ****
63 005B 928A827B    TABLE1: DEFB    92H,8AH,82H,7BH,75H
   005F 75
64 0060 6D6D6761            DEFB    6DH,6DH,67H,61H,5CH
   0064 5C
65 0065 57524D49            DEFB    57H,52H,4DH,49H
66 0069 4944413D    TABLE2: DEFB    49H,44H,41H,3DH,3AH
   006D 3A
67 006E 36363330            DEFB    36H,36H,33H,30H,2DH
   0072 2D
68 0073 2B282625            DEFB    2BH,28H,26H,25H
69 0077 2522201E    TABLE3: DEFB    25H,22H,20H,1EH,1CH
   007B 1C
70 007C 1A1A1918            DEFB    1AH,1AH,19H,18H,16H
   0080 16
71 0081 151413              DEFB    15H,14H,13H
72 0084                             END
```

3.6 音符データの入力演奏

音符データは、８４H番地以降の任意のアドレスに順に入力していきます。演奏の場合、アドレスを００００にし、DATAをその音符データの入力開始アドレスにしておきます。そして、RUNさせますと自動演奏が始まります。

（例）きよしこの夜（聖夜）

音	データ	音	データ	音	データ	音	データ
ソ	1A	ソ	1A	ソ	1A	ソ	1A
	06		0C		06		06
ラ	1C	ラ	1C	ラ	1C	ファ	18
	02		08		02		02
ソ	1A	ラ	1C	ソ	1A	レ	14
	04		04		04		04
ミ	16	ド	22	ミ	16	ド	12
	0C		06		0C		0C
ソ	1A	シ	1E	レ	24	休	80
	06		02		08		10
ラ	1C	ラ	1C	レ	24	くり返し	00
	02		04		04		
ソ	1A	ソ	1A	ファ	28		
	04		06		06		
ミ	16	ラ	1C	レ	24		
	0C		02		02		
レ	24	ソ	1A	シ	1E		
	08		04		04		
レ	24	ミ	16	ド	22		
	02		0C		0C		
シ	1E	ラ	1C	ミ	26		
	02		08		08		
シ	1E	ラ	1C	休	80		
	0C		04		04		
ド	22	ド	22	ド	22		
	08		06		04		
ド	22	シ	1E	ソ	1A		
	02		02		04		
ソ	1A	ラ	1C	ミ	16		
	02		04		04		

SM-B-80T
アプリケーションマニュアル(2)

13

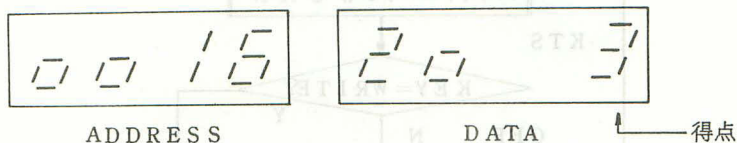
目　　次

1. スロット・マシン・ゲーム

1.1 概　　要

7セグメントのＬＥＤを利用したゲームのプログラムです。

プログラムのステップ数は，５４４バイトで，データ・エリアは，１６バイトです。

このゲームは，変化している６桁の表示を，できるだけ同じ数に揃えるように，タイミング良く［ＷＲＩＴＥ］キーを押して，得点を取るゲームです。

キーを押す毎に，２桁ずつ表示が固定され，３回で完了です。

得点の計算は，重複を許す組み合わせで行います。従って，例えば全部一致すれば，次のようになります。

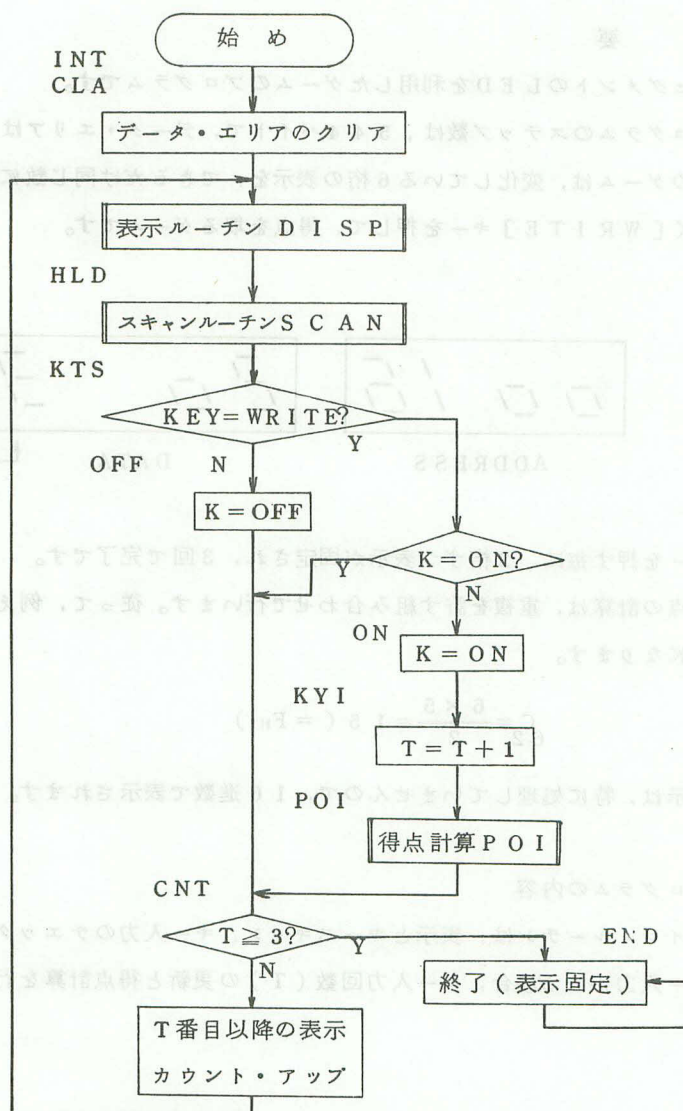
$$ {}_{6}C_{2} = \frac{6 \times 5}{2} = 15 \ (= F_H) $$

表示は，特に処理していませんので，１６進数で表示されます。

1.2 プログラムの内容

メイン・ルーチンは，表示とキースキャン，キー入力のチエックと表示データの更新です。

キー入力がある場合，キー入力回数（Ｔ）の更新と得点計算を行います。

1.2.1 フローチャート

始め
INT
CLA
データ・エリアのクリア
表示ルーチンDISP
HLD
スキャンルーチンSCAN
KTS
KEY=WRITE?
Y
OFF
N
K=OFF
K=ON?
Y
N
ON
K=ON
KYI
T=T+1
POI
得点計算POI
CNT
T≧3?
Y
END
N
終了(表示固定)
T番目以降の表示
カウント・アップ
○ 表示データ・エリア
DD 3 4 ADRS 右側表示データ
DD+1 1 2 ADRS 左側表示データ
DD+2 7 8 DATA 右側(ブランクと得点)
DD+3 5 6 DATA 左側表示データ
○ 比較用エリア
CD 1
CD+1 2
CD+2 3
CD+3 4
CD+4 5
CD+5 6

1.2.2　プログラムの詳細

キー入力がない間は，表示データをループ1周毎にインクリメントしてゆきます。キー入力があれば，入力回数カウンタＴをインクリメントし，この内容で表示データの更新箇所を制限してゆきます。

キーが押し続けられても，単発入力と見做せるように，Ｋフラグを設け，Ｋ＝ＯＦＦの時にキー入力があれば，有効とします。

キー入力毎に，カウンタＴの内容が更新され，得点計算のルーチンに入ります。

得点計算は，表示データを比較エリアに移してから行いますが，この比較は，Ｔの内容に従って，すでに固定されたデータ同志についてだけ行います。

一致があれば，得点カウンタＰの内容をインクリメントしてゆきます。この計算は，計算ルーチンに入る毎にやり直します。

これらの処理は，キー入力が3回入るまで続けられ，3回目で，終了のルーチンに入り，最終時の内容を表示し続けます。

これを解除する場合，［ＲＥＳＥＴ］キーを押します。

1.3　プログラムの使用ルーチンとエリア

このプログラムは，次のようなモニタのルーチンを使用しています。

ルーチン	エントリィ番地
ＤＩＳＰ	Ｅ２Ｆ７Ｈ
ＳＣＡＮ	Ｅ３７０Ｈ

また，変数のセーブ・エリアは，

ＦＦＯＯＨ　～　ＦＦＯＦＨ

です。

1.4　プログラム・リストを，5～7ページに示します。

1.5　プログラムの実行

次の順序でキーを押します。

1.6　プログラムの終了

［RESET］

```
SLOT    ASMBL'D BY  Z80  ASSEMBLER  REV-A.2 08,09,'78
                    TITLE    SLOT-GAME

001     SLOT-GAME

 1                          ;       SLOT-MACHINE GAME
 2                          ;
 3                          ;
 4                          ;       SLOT
 5                          ;
 6                          TITLE   SLOT-GAME
 7                          ORG     00H
 8  0000  3E00      INI     LD      A,0H
 9  0002  3204FF            LD      (P),A
10  0005  3200FF            LD      (K),A
11  0008  3205FF            LD      (T),A
12  000B  3E00      ENT     LD      A,0H
13  000D  0606              LD      B,6H
14  000F  210AFF            LD      HL,CD
15  0012  77        CLA     LD      (HL),A
16  0013  23                INC     HL
17  0014  10FC              DJNZ    CLA
18  0016  DD2106FF          LD      IX,DD
19  001A  ED4B06FF          LD      BC,(DD)
20  001E  ED43E1FF          LD      (AD),BC
21  0022  CD6E00            CALL    DSR
22  0025  060A              LD      B,AH
23  0027  C5        HLD     PUSH    BC
24  0028  CD70E3            CALL    SCAN
25  002B  C1                POP     BC
26  002C  10F9              DJNZ    HLD
27  002E  EE52      KTS     CP      52H
28  0030  2008              JR      NZ,OFF
29  0032  3A00FF            LD      A,(K)
30  0035  A7                AND     A
31  0036  2809              JR      Z,ON
32  0038  1813              JR      CNT
33  003A  3E00      OFF     LD      A,0H
34  003C  3200FF            LD      (K),A
35  003F  180C              JR      CNT
36  0041  3E01      ON      LD      A,1H
37  0043  3200FF            LD      (K),A
38  0046  2105FF    KYI     LD      HL,T
39  0049  34                INC     (HL)
40  004A  CD8300            CALL    POI
41  004D  3A05FF    CNT     LD      A,(T)
42  0050  A7                AND     A
43  0051  2808              JR      Z,IC0
44  0053  3D                DEC     A
45  0054  2808              JR      Z,IC1
46  0056  3D                DEC     A
47  0057  2808              JR      Z,IC2
48  0059  1808              JR      END
49  005B  DD3401    IC0     INC     (IX+1)
50  005E  DD3400    IC1     INC     (IX)
```

```
 51  0061  DD3403    IC2   INC   (IX+3)
 52  0064  18A5            JR    ENT
 53                 ;
 54  0066  CD6E00    END   CALL  DSR
 55  0069  CD70E3    ENDE  CALL  SCAN
 56  006C  18FB            JR    ENDE
 57                 ;
 58  006E  ED4B08FF  DSR   LD    BC,(DD+2)
 59  0072  3A04FF          LD    A,(P)
 60  0075  4F              LD    C,A
 61  0076  ED43DFFF        LD    (DT),BC
 62  007A  CDF7E2          CALL  DISP
 63  007D  3E00            LD    A,0H
 64  007F  32CEFF          LD    (SB1),A
 65  0082  C9              RET
 66                 ;
 67  0083  3A07FF    PO1   LD    A,(DD+1)
 68  0086  21DBFF          LD    HL,CD+1
 69  0089  ED67            RRD
 70  008B  320AFF          LD    (CD),A
 71  008E  3A06FF          LD    A,(DD)
 72  0091  210DFF          LD    HL,CD+3
 73  0094  ED67            RRD
 74  0096  320CFF          LD    (CD+2),A
 75  0099  3A09FF          LD    A,(DD+3)
 76  009C  210FFF          LD    HL,CD+5
 77  009F  ED67            RRD
 78  00A1  320EFF          LD    (CD+4),A
 79  00A4  3E00      TST   LD    A,0H
 80  00A6  3204FF          LD    (P),A
 81  00A9  2104FF          LD    HL,P
 82  00AC  3A05FF          LD    A,(T)
 83  00AF  EE00            CP    0H
 84  00B1  2840            JR    Z,TED
 85  00B3  FE01            CP    1H
 86  00B5  2835            JR    Z,T1
 87  00B7  FE02            CP    2H
 88  00B9  2821            JR    Z,T2
 89  00BB  FE03            CP    3H
 90  00BD  2802            JR    Z,T3
 91  00BF  1832            JR    TED
 92  00C1  ED4B0AFF  T3    LD    BC,(CD)
 93  00C5  ED5B0EFF        LD    DE,(CD+4)
 94  00C9  CDFA00          CALL  MID
 95  00CC  ED4B0CFF        LD    BC,(CD+2)
 96  00D0  ED5B0EFF        LD    DE,(CD+4)
 97  00D4  CDFA00          CALL  MID
 98  00D7  42              LD    B,D
 99  00D8  4B              LD    C,E
100  00D9  CDF400          CALL  THN
```

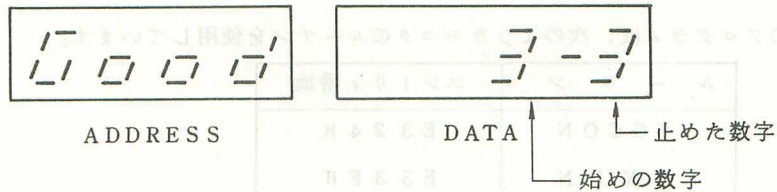
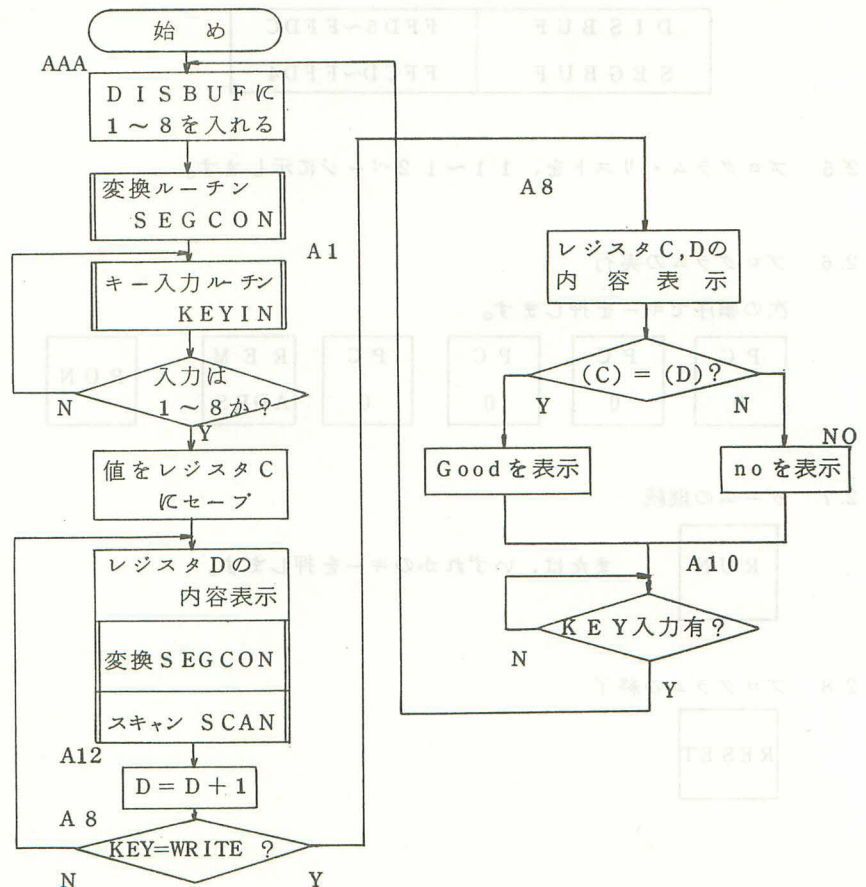
```
101 00DC ED4B0AFF T2   LD   BC,(CD)
102 00E0 ED5B0CFF      LD   DE,(CD+2)
103 00E4 CDFA00        CALL MID
104 00E7 42            LD   B,D
105 00E8 4B            LD   C,E
106 00E9 CDF400        CALL THN
107 00EC ED4B0AFF T1   LD   BC,(CD)
108 00E0 CDF400        CALL THN
109 00F3 C9       TED  RET
110               ;
111 00F4 78       THN  LD   A,B
112 00F5 B9            CP   C
113 00F6 2001          JR   NZ,THE
114 00F8 34            INC  (HL)
115 00F9 C9       THE  RET
116               ;
117 00FA 78       MID  LD   A,B
118 00FB C5            PUSH BC
119 00FC CD0501        CALL DEP
120 00FF C1            POP  BC
121 0100 79            LD   A,C
122 0101 CD0501        CALL DEP
123 0104 C9            RET
124               ;
125 0105 BA       DEP  CP   D
126 0106 2001          JR   NZ,DEA
127 0108 34            INC  (HL)
128 0109 BB       DEA  CP   E
129 010A 2001          JR   NZ,DEE
130 010C 34            INC  (HL)
131 010D C9       DEE  RET
132               ;
133      FF00     K    EQU  FF00H
134      FF04     P    EQU  FF04H
135      FF05     T    EQU  FF05H
136      FF06     DD   EQU  FF06H
137      FF0A     CD   EQU  FF0AH
138               ;
139      FFE1     AD   EQU  FFE1H
140      FFDF     DT   EQU  FFDFH
141      FFCE     SB1  EQU  FFCEH
142      E2F7     DISP EQU  E2F7H
143      E370     SCAN EQU  E370H
144      E2B7     WAIT EQU  E2B7H
145 010E               END
```

2. 電子ルーレット

2.1 概　　要

7セグメントのＬＥＤを利用したゲーム・プログラムです。

プログラムのステップ数は，２５６バイトで，データ・エリアは，モニタ用ＲＡＭ内の表示データ・エリアを兼用しています。

始めに指定した数字と合うように，動いてゆく数字をタイミング良く［ＷＲＩＴＥ］ＫＥＹを押して，止めるゲームです。

2.2 フローチャート

2.3　プログラムの詳細

表示用のデータ・エリアは，モニタが使用しているエリア"ＤＩＳＢＵＦ"を利用します。

始めのキー入力データは，レジスタＣに置き，止めた時のキー入力データは，レジスタＤに置きます。これらのレジスタの内容が壊れる恐れのある場合，スタックへＰＵＳＨしてゆきます。

このスタックのトップ・アドレスは，モニタ起動時に，ＦＦ９ＡＨ番地に設定されていますので，このプログラム内では設定せずに使っています。

2.4　プログラムの使用ルーチン

このプログラムは，次のようなモニタのルーチンを使用しています。

ルーチン	エントリィ番地
ＳＥＧＣＯＮ	Ｅ３２４Ｈ
ＫＥＹＩＮ	Ｅ３３ＦＨ
ＳＣＡＮ	Ｅ３７０Ｈ

表示用エリアは，次の２箇所で詳細はユーザーズ・マニュアルを御覧下さい。

ＤＩＳＢＵＦ	ＦＦＤ５～ＦＦＤＣ
ＳＥＧＢＵＦ	ＦＦＣＤ～ＦＦＤ４

2.5　プログラム・リストを，１１～１２ページに示します。

2.6　プログラムの実行

次の順序でキーを押します。

2.7　ゲームの継続

または，いずれかのキーを押します。

2.8　プログラムの終了

ROULETTE ASMBL'D BY Z80 ASSEMBLER REV—A.2 08,09,'78

TITLE ROULETTE

001   ROULETTE

```
 1                      ;        ELECTRONIC ROULETTE
 2                      ;
 3                               TITLE ROULETTE
 4                               ORG     00H
 5  0000  21D5FF        AAA:     LD      HL,DISBUF
 6  0003  0608                   LD      B,8H
 7  0005  70            LOOP:    LD      (HL),B
 8  0006  23                     INC     HL
 9  0007  10FC                   DJNZ    LOOP
10  0009  CD24E3                 CALL    SEGCON
11  000C  CD3FE3        A1:      CALL    KEYIN
12  000F  A7                     AND     A
13  0010  28FA                   JR      Z,A1
14  0012  FE09                   CP      9
15  0014  30F6                   JR      NC,A1
16  0016  4F                     LD      C,A
17  0017  0608                   LD      B,8
18  0019  3E01                   LD      A,1
19  001B  57                     LD      D,A
20  001C  21DCFF        A2:      LD      HL,DIS
21  001F  0F            A3:      RRCA
22  0020  3804                   JR      C,A4
23  0022  3617                   LD      (HL),17H
24  0024  1801                   JR      A5
25  0026  72            A4:      LD      (HL),D
26  0027  2B            A5:      DEC     HL
27  0028  10F5                   DJNZ    A3
28  002A  07                     RLCA
29  002B  3002                   JR      HC,A6
30  002D  1600                   LD      D,0
31  002F  F5            A6:      PUSH    AF
32  0030  061F                   LD      B,1FH
33  0032  E5            A7:      PUSH    HL
34  0033  D5                     PUSH    DE
35  0034  C5                     PUSH    BC
36  0035  CD24E3                 CALL    SEGCON
37  0038  CD70E3                 CALL    SCAN
38  003B  C1                     POP     BC
39  003C  D1                     POP     DE
40  003D  E1                     POP     HL
41  003E  2006                   JR      NZ,A8
42  0040  10F0          A12:     DJNZ    A7
43  0042  F1                     POP     AF
44  0043  14                     JNC     D
45  0044  18D6                   JR      A2
46  0046  FE52          A8:      CP      52H
47  0048  20F6                   JR      NZ,A12
48  004A  F1                     POP     AF
49  004B  7A                     LD      A,D
50  004C  B9                     CP      C
```

```
51  004D  F5                          PUSH   AF
52  004E  21D5FF                      LD     HL,DISBUF
53  0051  77                          LD     (HL),A
54  0052  23                          INC    HL
55  0053  3615                        LD     (HL),15H
56  0055  23                          INC    HL
57  0056  71                          LD     (HL),C
58  0058  CD24E3                      CALL   SEGCON
59  005A  F1                          POP    AF
60  005B  21D4FF                      LD     HL,SEGBUF+7
61  005E  2017                        JR     NZ,NO
62  0060  363D                        LD     (HL),3DH
63  0062  2B                          DEC    HL
64  0063  365C                        LD     (HL),5CH
65  0065  2B                          DEC    HL
66  0066  365C                        LD     (HL),5CH
67  0068  2B                          DEC    HL
68  0069  365E                        LD     (HL),5EH
69  006B  0601          A9:           LD     B,1
70  006D  2B            A10:          DEC    HL
71  006E  3600                        LD     (HL),0
72  0070  10FB                        DJNZ   A10
73  0072  CD3FE3                      CALL   KEYIN
74  0075  1889                        JR     AAA
75  0077  3654          NO:           LD     (HL),54H
76  0079  2B                          DEC    HL
77  007A  365C                        LD     (HL),5CH
78  007C  0603                        LD     B,3
79  007E  18ED                        JR     A10
80        FFDC          DIS           EQU    FFD5H+7
81        E324          SEGCON        EQU    E324H
82        FFD5          DISBUF        EQU    FFD5H
83        E370          SCAN          EQU    E370H
84        E33F          KEYIN         EQU    E33FH
85        FFCD          SEGBUF        EQU    FFCDH
86  0080                              END
```

3. 紙テープリーダ・インターフェース

3.1 概　　要

ＳＭ－Ｂ－８０Ｔのメモリへ紙テープからデータ（オブジェクトプログラム）を入力するプログラムです。

このプログラムは，紙テープリーダにリコー電子工業株式会社製のＭｉ，ＲＥＡＤＥＲを使用し，この紙テープリーダとのインターフェースにＳＭ－Ｂ－８０Ｔの汎用パラレルＩ／Ｏ（ＰＩＯ２）を使用しています。また，プログラムの開始アドレスは，ユーザＲＯＭ領域のＥ４００番地とします。

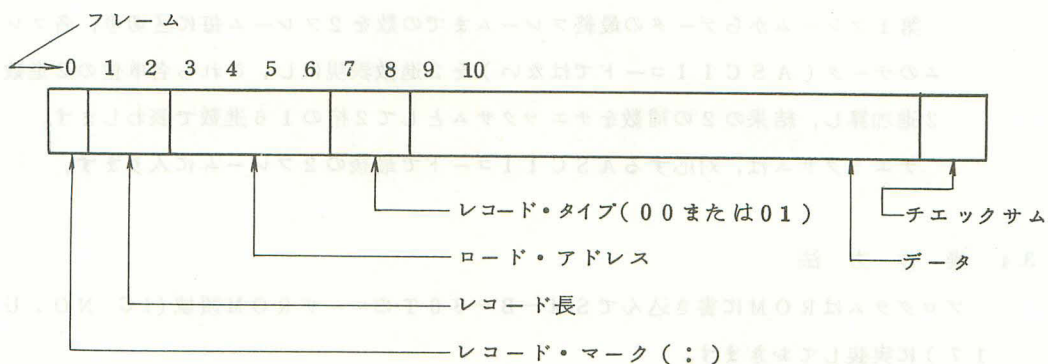
3.2 プログラム

リーダ・ドライバールーチンのリストを，後で示します。

このプログラムは，Ｅ４００Ｈ番地からのＲＯＭエリアに書き込んで使うようにしてありますが，他のメモリ・エリアに置いて使うこともできます。

但し，この場合，ＣＡＬＬ命令及びＪＰ命令内のオペランド（ジャンプ先の番地）はすべて修正する必要があるので，御注意下さい。

3.3 データの形式

データ（オブジェクトプログラム）は，インテル標準１６進フォーマットとします。

3.3.1 レコード・マーク

コロン（：）に対応するＡＳＣＩＩコード３Ａによりレコードの先頭を示します。

3.3.2 レコード長

レコードのデータ（バイト）長をＡＳＣＩＩコードで表わします。例えば，データ長１２９バイトは１６進数で８１であり，ＡＳＣＩＩコードでは３８３１となります。

エンド・オフ・ファイルのレコード長は００であり，フレーム１，２には対応するＡＳＣＩＩコードが入ります。

3.3.3　ロード・アドレス

レコードの先頭データがロードされるアドレスをＡＳＣＩＩコードで表わします。アドレスは４桁の１６進数で表わし，上位側がフレームの３，４に入ります。

エンド・オフ・ファイルのアドレスは００００であり，フレーム３～６に対応するＡＳＣＩＩコードが入ります。

3.3.4　レコード・タイプ

データ・レコードはタイプ１，エンド・オフ・ファイルはタイプ０として各々１，０に対応するＡＳＣＩＩコードで表わします。

3.3.5　データ

１バイト・データは２桁の１６進数で表現できますが，これに対応するＡＳＣＩＩコードで連続する２フレームに入れます。データの上位バイトが前のフレーム，下位バイトが後のフレームになります。

3.3.6　チエックサム

第１フレームからデータの最終フレームまでの数を２フレーム毎に区切り，各フレームのデータ（ＡＳＣＩＩコードではない）を２進数表現にし，これら各単位の２進数を２進加算し，結果の２の補数をチエックサムとして２桁の１６進数で表わします。

チエックサムは，対応するＡＳＣＩＩコードで最後の２フレームに入ります。

3.4　操　作　方　法

プログラムはＲＯＭに書き込んでＳＭ－Ｂ－８０ＴのユーザＲＯＭ領域（ＩＣ　ＮＯ．Ｕ１７）に実装しておきます。

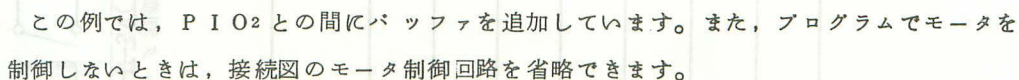
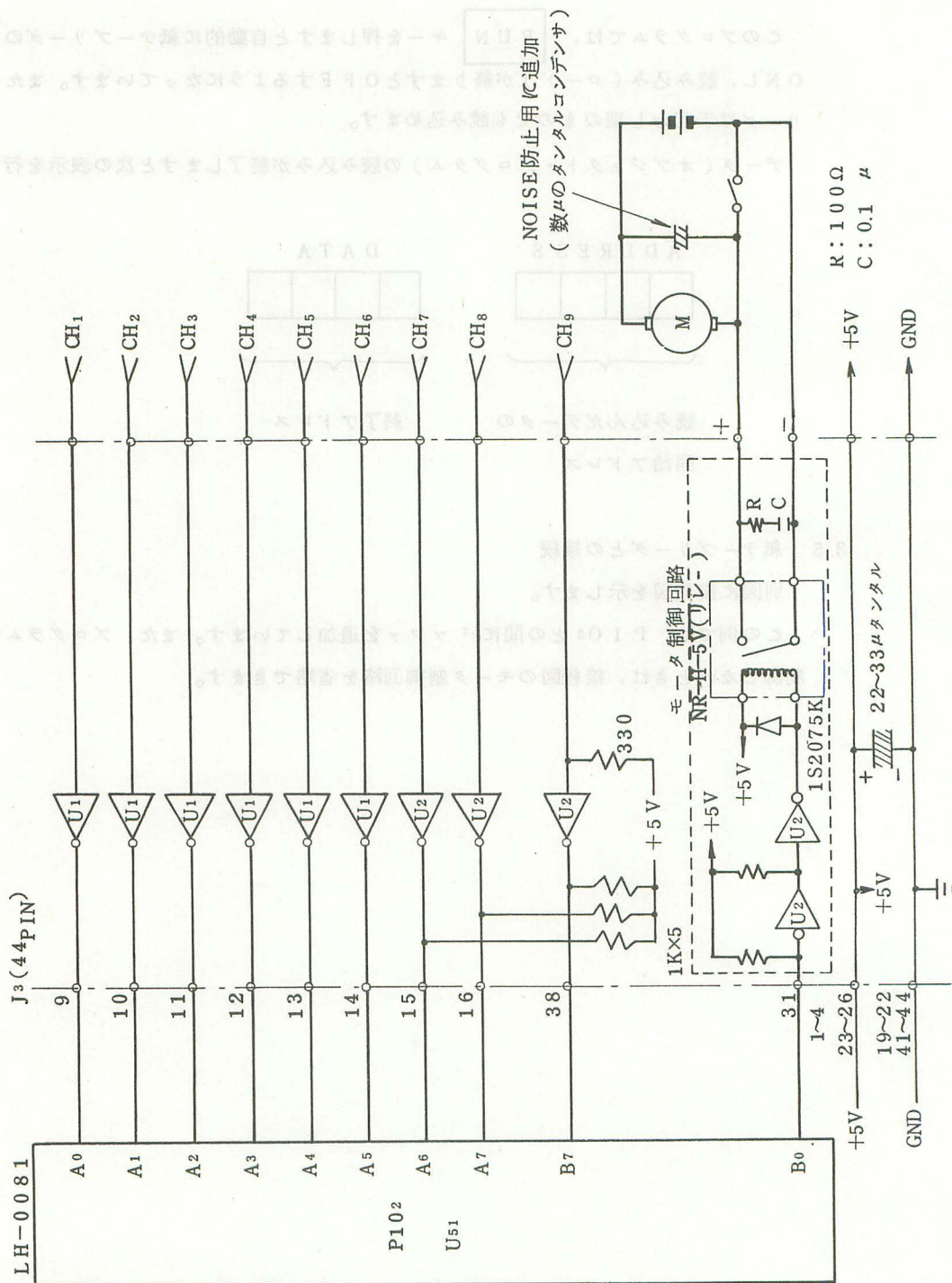
キー操作

このプログラムでは，[RUN]キーを押しますと自動的に紙テープリーダのモータをONし，読み込み（ロード）が終りますとOFFするようになっています。また，紙テープリーダは手回わし型のものでも読み込めます。

データ（オブジェクト・プログラム）の読み込みが終了しますと次の表示を行います。

3.5 紙テープリーダとの接続

別図に接続図を示します。

この例では，PIO2との間にバッファを追加しています。また，プログラムでモータを制御しないときは，接続図のモータ制御回路を省略できます。

SM−B−80T，MiREADER 接続図

U1 7404

U2 7417

```
001 LOAD. SR
  1                        ;
  2                        ;
  3                        ;           ★★★★★ LOAD ★★★★★
  4                        ;
  5        00D4          PIOAD     EQU     0D4H
  6        00D5          PIOAC     EQU     0D5H
  7        00D6          PIOBD     EQU     0D6H
  8        00D7          PIOBC     EQU     0D7H
  9        E026          EXIT      EQU     0E026H
 10        FFE1          ADDR      EQU     0FFE1H
 11        FFDF          DATA      EQU     0FFDFH
 12        FFCD          SEGBUF    EQU     0FFCDH
 13                                ORG     0E400H
 14  E400  3ECF          LOAD:     LD      A,0CFH
 15  E402  D3D5                    OUT     (PIOAC),A
 16  E404  3EFF                    LD      A,0FFH
 17  E406  D3D5                    OUT     (PIOAC),A
 18  E408  3ECF                    LD      A,0CFH
 19  E40A  D3D7                    OUT     (PIOBC),A
 20  E40C  3EFE                    LD      A,0FEH
 21  E40E  D3D7                    OUT     (PIOBC),A
 22  E410  AF                      XOR     A
 23  E411  D3D6                    OUT     (PIOBD),A
 24  E413  1E00                    LD      E,0
 25  E415  CD82E4        LD20:     CALL    READ
 26  E418  FE3A                    CP      ':'
 27  E41A  20F9                    JR      NZ,LD20
 28  E41C  0E00                    LD      C,0
 29  E41E  CD61E4                  CALL    RDBYT
 30  E421  A7                      AND     A
 31  E422  2836                    JR      Z,LD100
 32  E424  47                      LD      B,A
 33  E425  CD61E4                  CALL    RDBYT
 34  E428  67                      LD      H,A
 35  E429  CD61E4                  CALL    RDBYT
 36  E42C  6F                      LD      L,A
 37  E42D  CD61E4                  CALL    RDBYT
 38  E430  CB43                    BIT     0,E
 39  E432  2004                    JR      NZ,LD30
 40  E434  22E1FF                  LD      (ADDR),HL
 41  E437  1C                      INC     E
 42  E438  CD61E4        LD30:     CALL    RDBYT
 43  E43B  77            LD50:     LD      (HL),A
 44  E43C  22DFFF                  LD      (DATA),HL
 45  E43F  23            LD60:     INC     HL
 46  E440  10F6                    DJNZ    LD30
 47  E442  CD61E4                  CALL    RDBYT
 48  E445  AF                      XOR     A
 49  E446  81                      ADD     A,C
 50  E447  28CC                    JR      Z,LD20
```

```
002      LOAD.SR
   51 E449 0608              LD      B,8
   52 E44B 21CDFF            LD      HL,SEGBUF
   53 E44E 3640              LD      (HL),40H
   54 E450 23                INC     HL
   55 E451 10FB              DJNZ    $-3
   56 E453 3E01              LD      A,1
   57 E455 D3D6              OUT     (PIOBD),A
   58 E457 C32CE0            JP      EXIT+6
   59 E45A 3E01     LD100:   LD      A,1
   60 E45C D3D6              OUT     (PIOBD),A
   61 E45E C326E0            JP      EXIT
   62 E461 CD82E4   RDBYT:   CALL    READ
   63 E464 CD7AE4            CALL    A2BIN
   64 E467 07                RLCA
   65 E468 07                RLCA
   66 E469 07                RLCA
   67 E46A 07                RLCA
   68 E46B C5                PUSH    BC
   69 E46C 4F                LD      C,A
   70 E46D CD82E4            CALL    READ
   71 E470 CD7AE4            CALL    A2BIN
   72 E473 B1                OR      C
   73 E474 C1                POP     BC
   74 E475 F5                PUSH    AF
   75 E476 81                ADD     A,C
   76 E477 4F                LD      C,A
   77 E478 F1                POP     AF
   78 E479 C9                RET
   79 E47A D630     A2BIN:   SUB     '0'
   80 E47C FE0A              CP      10
   81 E47E F8                RET     M
   82 E47F D607              SUB     7
   83 E481 C9                RET
   84 E482 DBD6     READ:    IN      A,(PIOBD)
   85 E484 07                RLCA
   86 E485 38FB              JR      C,READ
   87 E487 DBD4              IN      A,(PIOAD)
   88 E489 2F                CPL
   89 E48A E67F              AND     7FH
   90 E48C F5                PUSH    AF
   91 E48D DBD6     RD10:    IN      A,(PIOBD)
   92 E48F 07                RLCA
   93 E490 30FB              JR      NC,RD10
   94 E492 F1                POP     AF
   95 E493 C9                RET
   96 E494                   END
```

LOAD.SR SYMBOL TABLE

```
PIOAD -00D4    PIOAC -00D5    PIOBD -00D6    PIOBC -00D7
EXIT  -E026    ADDR  -FFE1    DATA  -FFDF    SEGBUF-FFCD
LOAD  -E400    LD20  -E415    LD30  -E438    LD50  -E43B
LD60  -E43F    LD100 -E45A    RDBYT -E461    A2BIN -E47A
READ  -E482    RD10  -E48D
```

APPENDIX 14
ROM/RAMボード

目　　次

1. 概　　要

ROM／RAMボードは、SM－B－80DやSM－B－80Tを使用したシステムにおいて、メモリを拡張する場合に用いる。

このボードはROMとして、2708タイプ、または2716タイプのEPROMを最大8個まで実装できる。2708タイプのEPROMの最大実装容量は8KBであり、基準アドレスは4KB単位に変更できる。2716タイプのEPROMの最大実装容量は16KBであり、基準アドレスは8KB単位に変更できる。

RAMとして8KBのスタティックRAMを実装しており、その基準アドレスは4KB単位に変更できる。また、このRAMは必要に応じて4KB単位に書き込みを不可能にするメモリ・プロテクト機能を持っている。

1.1　特　　長

(1) RAM　8KB実装　LH－2114－3　16個使用。

(2) ROM　2708タイプ、intel 2716タイプ　いずれも使用可能。

(3) ROM容量　2708使用時　最大8KB
　　　　　　　2716使用時　最大16KB

(4) RAM基準アドレスは4KB単位に変更できる。

(5) ROM基準アドレスは、2708使用時4KB、2716使用時8KB単位に変更できる。

(6) RAMは4KB単位に書き込みを不可能にできる（メモリ・プロテクト機能）。

1.2　仕　　様

＜一般仕様＞

項　　目	仕　　　　様	備　　　　考
RAM	8KB 実装 LH－2114－3　16個使用 4KB単位に基準アドレス設定可能 4KB単位にメモリ・プロテクト可能	 DIPスイッチにより設定
ROM	2708または2716を最大8個まで実装できる。 ROM容量 　2708使用時　最大8KB 　2716使用時　最大16KB ROM基準アドレス 　2708使用時　4KB単位可変 　2716使用時　8KB単位可変	ボードはICソケットのみ実装している。 2716はintel compatibleのEPROM DIPスイッチにより設定
電　　源	＋5V±5%；2Amax＋Icc(EPROM) ＋12V±5%；IDD(EPROM) －12V±5%；IBB(EPROM) (－5V±5%)	 ＋12V，－12V(－5V)はEPROM使用時必要 －12Vの代わりに－5Vを直接印加することもできる。
動作温度	0℃～40℃	
ボード寸法	270×190×20　　　単位mm	
バス信号コネクタ	100ピン　3.175mmピッチ	

（注意）(1)　2708タイプ、2716タイプのEPROMの混用はできない。

(2)　2708タイプのEPROMの代わりに端子互換性のあるバイポーラPROM（例LH－7055）を使用できる。

(3)　電源項目のIcc，IDD，IBBはEPROMの電源供給電流。EPROMの使用個数nにより、この値はn倍される。

＜バス信号線＞

ＲＯＭ／ＲＡＭボードに入出力するバス信号線の仕様を以下に示す。

(1) 信号の論理

データ　　負論理

アドレス　　負論理

制御線　　負論理

(2) 信号のレベル

データ　　ＴＴＬコンパティブル

（双方向バッファ８Ｔ２６使用）

アドレス　　低電力ＴＴＬコンパティブル

制御線　　低電力ＴＴＬコンパティブル

（但し、＊ＲＥＮＢのみ負荷１ＫΩによるオープン・コレクタ出力）

バス信号線の一覧表を表１に示す。

＜ボード寸法図＞

図１にボード寸法図を示す。

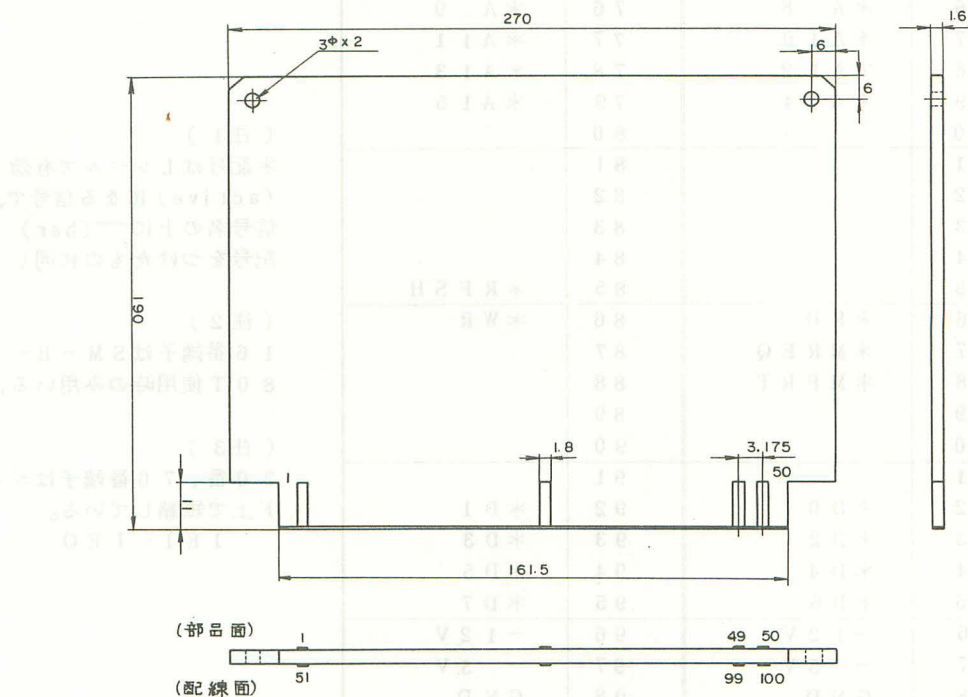



図1 ボード寸法図

端子番号	信号名（部品面）	端子番号	信号名（配線面）
1	＋5V	51	＋5V
2	＋5V	52	＋5V
3	＋5V	53	＋5V
4		54	
5	＋12V	55	＋12V
6		56	
7		57	
8		58	
9		59	
10		60	
11		61	
12		62	
13		63	
14		64	
15		65	＊RENB（注1）
16	＊RENB（注2）	66	
17		67	
18		68	
19		69	
20	（注3）	70	
21		71	
22	＊A0	72	＊A1
23	＊A2	73	＊A3
24	＊A4	74	＊A5
25	＊A6	75	＊A7
26	＊A8	76	＊A9
27	＊A10	77	＊A11
28	＊A12	78	＊A13
29	＊A14	79	＊A15
30		80	
31		81	
32		82	
33		83	
34		84	
35		85	＊RFSH
36	＊RD	86	＊WR
37	＊MREQ	87	
38	＊MPRT	88	
39		89	
40		90	
41		91	
42	＊D0	92	＊D1
43	＊D2	93	＊D3
44	＊D4	94	＊D5
45	＊D6	95	＊D7
46	－12V	96	－12V
47	－5V	97	－5V
48	GND	98	GND
49	GND	99	GND
50	GND	100	GND

（注1）
＊記号はLレベルで有効(active)になる信号で、信号名の上に―(bar)記号をつけたものに同じ

（注2）
16番端子はSM－B－80T使用時のみ用いる。

（注3）
20番，70番端子はボード上で短絡している。
IEI＝IEO

2. ハードウエア

2.1　回路説明

図2にROM／RAMボードのブロック図を示す。また、本マニュアルの最後に部品構成表、回路図を示す。

(1) アドレス・バッファ

U_{38}～U_{40}のインバータにより＊A_0～＊A_{15}のアドレス入力をバッファする。

(2) アドレス・デコーダ

アドレス・デコーダは、RAM，ROMに対して別々に設けられており、RAMに対する基準アドレスはDIPスイッチDP2により、ROMに対する基準アドレスはDP3により設定できる。U_{36}はRAMに対するアドレス・デコーダであり、1KB単位にRAMを選択している。U_{37}はROMに対するアドレス・デコーダであり、1KB単位または2KB単位にROMを選択している。

(3) データ・バッファ

U_{17}，U_{18}の双方向バッファによりデータを入出力する。双方向バッファのレシーバ、及びドライバがイネーブルとなる条件を次に示す。

$$DE = \overline{*RD + (*MS1)\cdot(*MS2)\cdot(*MS3)\cdot(*MS4)}$$

$$\overline{RE} = *WR + (*MS1)\cdot(*MS2)\cdot(*MS3)\cdot(*MS4)$$

但し、MSiが"L"になる条件は、ボード上のメモリが選択され、かつ、＊MREQが"L"レベルで、＊RFSHが"H"レベルとなる場合である。

(4) RAMアレー

U_9～U_{16}，U_{22}～U_{29}は、LH－2114－3、スタティックRAMアレーである。U_9～U_{12}，U_{22}～U_{25}及びU_{13}～U_{16}，U_{26}～U_{29}により各々4KBのRAMブロック(1)，(2)を構成している。

(5) ROMアレー

U_1～U_8は2708，または2716のROMアレーが構成できるようになったICソケットであり、2708，2716の使用切り換えはジャンパ線P_1～P_{10}及びDIPスイッチDP1(6)により行う。詳細は3.使用法を参照のこと。

(6) メモリ・プロテクト回路

メモリ・プロテクトは上述の4KB単位のRAMブロックに対して書き込みを不可能にするものであり書き込みが可能となる条件は以下の通りである。

38番端子＊MPRTが"L"レベルで、かつ、DIPスイッチDP1の(7)，(8)がオンの場合

2.2　バス信号線

表2にバス信号線の意味を示す。

(注) ROM ブロック(1).(2)はソケットのみ実装

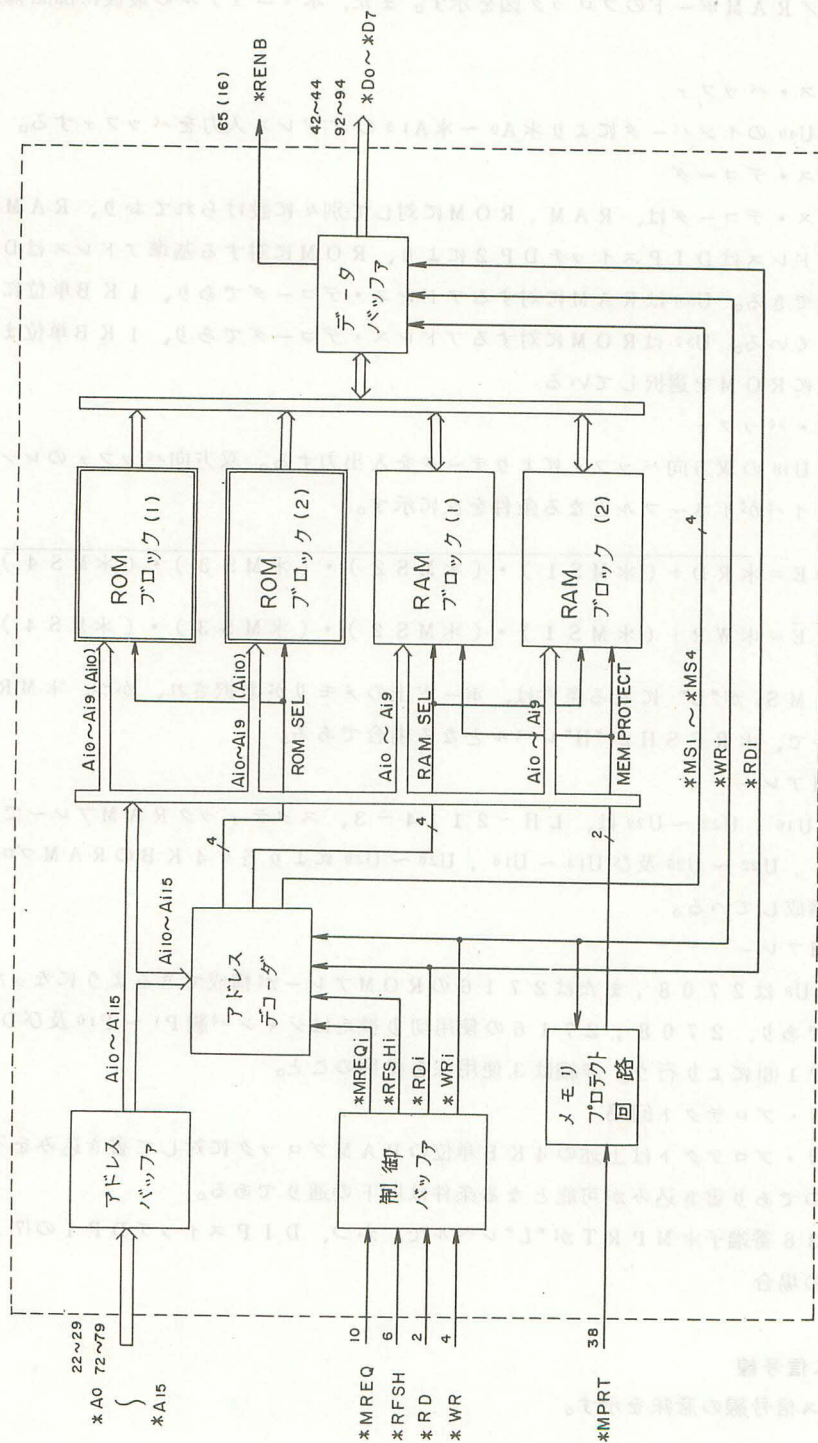



図 2. ブロック図

表2　バス信号線の意味

信号名	端子番号	説明	信号方向
＋５Ｖ	1,2,3 51,52,53	電源＋５Ｖ，i２７１６またはＬＨ７０５５をＲＯＭとして使用する場合＋５Ｖだけでよい。	入力
＋１２Ｖ	５，５５	電源＋１２Ｖ，２７０８タイプのＥＰＲＯＭを使用する場合本電源を必要とする。	入力
＊ＲＥＮＢ	６５ 又は １６	Receiver ENaBle この端子を"L"レベルにすることにより、ＣＰＵボードのデータ・バッファのレシーバはイネーブルになる。本ボード上のメモリをアクセスすると本端子は"L"レベルになる。	出力 "L"レベルアクティブ
アドレス ＊A_0 ～ ＊A_{15}	右参照	Address アドレス入力端子であり、本端子に印加されたアドレス信号は反転バッファ後、ボード上のアドレス・デコーダ、メモリのアドレス入力に印加される。 記号　*A0 *A1 *A2 *A3 *A4 *A5 *A6 *A7 *A8 *A9 *A10 *A11 *A12 *A13 *A14 *A15 端子　22 72 23 73 24 74 25 75 26 76 27 77 28 78 29 79	入力 "L"レベルアクティブ
＊ＲＦＳＨ	８５	Re Fre SH ダイナミックＲＡＭのリフレッシュ・サイクルの期間本信号はアクティブになる。本ボードはダイナミックＲＡＭを使用していないが、リフレッシュ・アドレスによってボード上のメモリがアクセスされるのを、本信号を用いて禁止する。	入力 "L"レベルアクティブ
＊ＲＤ	３６	Re aD ＣＰＵの読み出しサイクルの期間本信号はアクティブとなる。	入力 "L"レベルアクティブ
＊ＷＲ	８６	W Rite ＣＰＵの書き込みサイクルの期間本信号はアクティブとなる。メモリ・プロテクトにより本信号は無効になる。	入力 "L"レベルアクティブ
＊ＭＲＥＱ	３７	Memory REQuest メモリに対する読み出し／書き込み、リフレッシュ・サイクルの期間、本信号はアクティブとなる。	入力 "L"レベルアクティブ

信　号　名	端子番号	説　　　明	信号方向
データ *D0 〜 *D7	右参照	Data データの入出力端子であり、双方向バッファ8T26でバッファ後、ボード上のメモリのデータ端子に接続している。双方向バッファのドライバがアクティブになるのは、＊MREQが"L"レベル、＊RFSHが"H"レベルで、ボード上のメモリが選択され、かつ＊RDが"L"レベルとなる場合である。また、双方向バッファのレシーバがアクティブとなるのは、＊MREQが"L"レベル、＊RFSHが"H"レベルでボード上のメモリが選択され、かつ、＊WRが"L"レベルとなる場合である。 記号　*D0 *D1 *D2 *D3 *D4 *D5 *D6 *D7 端子　42 92 43 93 44 94 45 95	入出力 "L"レベル アクティブ
＊MPRT	38	Memory PRoTect 本信号を"L"レベルにし、かつ、プロテクト・スイッチ（DP1の(7)か(8)）がオンの場合、対応するRAMブロックに対して書き込みができない。	入力 "L"レベル アクティブ
−12V	46，96	電源−12V 2708タイプのEPROMの−5V電源は本端子に−12Vを供給することにより、ボード内部で作成できる。	入力
（−5V）	47，97	電源−5V 2708タイプのEPROMに直接−5V電源を供給する場合に使用する。	入力
GND	48,49,50 98,99,100	接　地　線	

3. 使　用　法

ROM／RAMボードは、2708，2716のEPROMの切り換え、メモリの基準アドレスの設定等をボード上のDIPスイッチ、チエック端子間の配線により行っている。表3はDIPスイッチ、チエック端子の各機能の一覧表である。

表3　動作モード切り換え一覧表

	番　号	機　　能
DIP スイッチ DP1	(1)	RAMブロック1に対して＊MREQを有効にする
	(2)	〃　〃　2　〃　〃
	(3)	ROM　〃　1　〃　〃
	(4)	〃　〃　2　〃　〃
	(5)	未使用
	(6)	2708時スイッチオン、2716時スイッチオフ
	(7)	RAMブロック1に対して書き込み不可能(プロテクト・オン)(注2)
	(8)	〃　〃　2　〃　〃　(　〃　)
DIP スイッチ DP2	(1)	RAM　ブロック1の基準アドレス設定　(A_{15})
	(2)	〃　〃　〃　(A_{14})
	(3)	〃　〃　〃　(A_{13})
	(4)	〃　〃　〃　(A_{12})
	(5)	RAM　ブロック2の基準アドレス設定　(A_{15})
	(6)	〃　〃　〃　(A_{14})
	(7)	〃　〃　〃　(A_{13})
	(8)	〃　〃　〃　(A_{12})
DIP スイッチ DP3	(1)	ROM　ブロック1の基準アドレス設定　(A_{15})
	(2)	〃　〃　〃　(A_{14})
	(3)	〃　〃　〃　(A_{13})
	(4)	〃　〃　〃　(A_{12})（注1）
	(5)	ROM　ブロック2の基準アドレス設定　(A_{15})
	(6)	〃　〃　〃　(A_{14})
	(7)	〃　〃　〃　(A_{13})
	(8)	〃　〃　〃　(A_{12})（注1）
チエック端子	P_1 P_2	P_1 − P_2 は通常結線、EPROMのVDD(A_{10})端子を開放にする場合、P_1 − P_2 を開放にする。
	P_3 P_4	P_3 − P_4 を結線すると −12Vからレギュレータを介して−5V をEPROMに供給できる。−5Vを直接供給する場合開放とする。
	P_5 P_6 P_7	P5 − P6 結線時EPROMの19番端子に A_{10} を印加 P7 − P6　〃　〃　〃　+12V　〃 （但しP1 − P2は結線とする）
	P_8 P_9 P_{10}	P8 − P9結線時EPROMの21番端子に＋5Vを印加 P10 − P9　〃　〃　〃　−5V　〃

（注1）　2716使用時(1)はオフとする。

（注2）　書き込み不可能となるには、さらに＊MPRT＝"L"が必要

DIP スイッチ

3.1　ＲＡＭ使用時の設定

(1)　ＤＩＰスイッチＤＰ１の(1)，(2)をオンにする。

(1)，(2)をオフにすると＊ＭＲＥＱが無効になり、対応するＲＡＭブロックがアクセスできなくなる。ボード上のメモリを使用しない場合、オフにする。

(2)　再書き込みを禁止したければ、前もってＤＰ１の(7)，(8)をオンにしておく。

(7)，(8)をオンにした後、＊ＭＰＲＴを"L"レベルにすると、ＲＡＭに対して以後書き込みはできなくなる（読み出しは可能）。

従って通常の使用法ではＤＰ１の(1)，(2)をオン、(7)，(8)をオフにする。

(3)　ＲＡＭの基準アドレスを表４に従って設定する。設定はＤＰ２により行う。

	ＲＡＭブロック１				ＲＡＭブロック２			
メモリ番号	U9～U12（上位ハーフバイト） U22～U25（下位ハーフバイト）				U13～U16（上位ハーフバイト） U26～U29（下位ハーフバイト）			
DP2番号 / 基準アドレス	1	2	3	4	5	6	7	8
００００	0	0	0	0	0	0	0	0
１０００	0	0	0	1	0	0	0	1
２０００	0	0	1	0	0	0	1	0
３０００	0	0	1	1	0	0	1	1
４０００	0	1	0	0	0	1	0	0
５０００	0	1	0	1	0	1	0	1
６０００	0	1	1	0	0	1	1	0
７０００	0	1	1	1	0	1	1	1
８０００	1	0	0	0	1	0	0	0
９０００	1	0	0	1	1	0	0	1
Ａ０００	1	0	1	0	1	0	1	0
Ｂ０００	1	0	1	1	1	0	1	1
Ｃ０００	1	1	0	0	1	1	0	0
Ｄ０００	1	1	0	1	1	1	0	1
Ｅ０００	1	1	1	0	1	1	1	0
Ｆ０００	1	1	1	1	1	1	1	1

1：対応スイッチオン
0：〃　〃オフ

表４　ＲＡＭ基準アドレスの設定

3.2　ROM使用時の設定

(1)　DIPスイッチDP1の(3)，(4)をオンにする。

(3)，(4)をオフにすると＊MREQが無効になり、対応するROMブロックがアクセスできなくなる。ボード上のメモリを使用しない場合オフとする。

(2)　2708使用時DP1の(6)をオンに、2716使用時DP1の(6)をオフにする。

(3)　ROMの基準アドレスを表5に従って設定する。設定はDP3により行う。

	ROMブロック1				ROMブロック2				
メモリ番号	U_1〜U_4				U_5〜U_8				
基準アドレス ＼ DP3番号	1	2	3	4	5	6	7	8	
0000	0	0	0	0	0	0	0	0	
1000	0	0	0	1	0	0	0	1	(※)
2000	0	0	1	0	0	0	1	0	
3000	0	0	1	1	0	0	1	1	(※)
4000	0	1	0	0	0	1	0	0	
5000	0	1	0	1	0	1	0	1	(※)
6000	0	1	1	0	0	1	1	0	
7000	0	1	1	1	0	1	1	1	(※)
8000	1	0	0	0	1	0	0	0	
9000	1	0	0	1	1	0	0	1	(※)
A000	1	0	1	0	1	0	1	0	
B000	1	0	1	1	1	0	1	1	(※)
C000	1	1	0	0	1	1	0	0	
D000	1	1	0	1	1	1	0	1	(※)
E000	1	1	1	0	1	1	1	0	
F000	1	1	1	1	1	1	1	1	(※)

1：対応スイッチオン
0：〃　〃　オフ

（注1）　2716の場合(※)印の基準アドレスは設定できない。従って2716使用時DP3の(4)，(8)スイッチは必ずオンとする。

（注2）　2708，2716の混用はできない。

表5　ROM基準アドレスの設定

(4)　2708，2716に対応してチェック端子を表6に従って設定する。

ROM ＼ チエック端子	EPROM 2708	EPROM 2716	PROM 7055	備　　考
P_1－P_2	1	1	0	
P_3－P_4	1(注)	0	0	(注) 直接－5V供給時は開放とする。
P_5－P_6	0	1	0	
P_7－P_6	1	0	0	
P_8－P_9	0	1	0	
P_{10}－P_9	1	0	0	

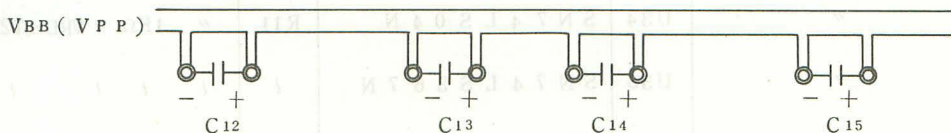
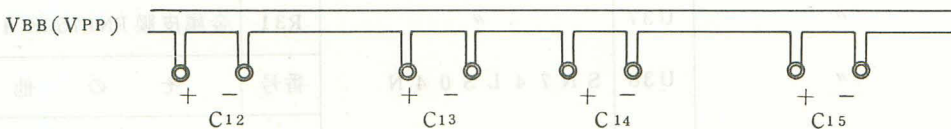
表6　チエック端子配線

1：結線
0：開放

(5)　2708と2716では21番端子の印加電圧の極性が異なるため2716使用時に次の変更を行う。

コンデンサC_{12}～C_{15}の実装方向を逆にする。

本ボードでは出荷時のコンデンサC_{12}～C_{15}の実装方向は2708用となっている。従って2716使用の場合、以下のように実装方向を逆にしなければならない。

・2708使用の場合

なお、変更に際しては、ボード添付の"シャープワンボードマイクロコンピュータシリーズご使用に当って"をよくお読み下さい。

3.3　SM－B－80Tの拡張メモリ

ROM／RAMボードをSM－B－80Tの拡張メモリとする場合、以下の変更を行わなければならない。

＊RENBを16番端子に出力する。

部品構成表

番号	ICソケット
U1	821D-24011-244-11
U2	〃
U3	〃
U4	〃
U5	〃
U6	〃
U7	〃
U8	〃

番号	IC
U9	LH－2114－3
U10	〃
U11	〃
U12	〃
U13	〃
U14	〃
U15	〃
U16	〃
U17	MC8T26AP
U18	〃
U19	SN75452BP
U20	SN74LS32N
U21	〃
U22	LH－2114－3
U23	〃

番号	IC
U24	LH－2114－3
U25	〃
U26	〃
U27	〃
U28	〃
U29	〃
U30	SN74LS367N
U31	SN74LS32N
U32	SN74LS02N
U33	SN74LS00N
U34	SN74LS04N
U35	SN74LS367N
U36	SN74LS139N
U37	〃
U38	SN74LS04N
U39	〃
U40	〃
U41	SN74LS266N
U42	〃
U43	〃
U44	〃

番号	コンデンサ
C1	電解47uF SL25VBSN47
C2	〃
C3	電解47uF SL25VBSN47
C4	タンタル22uF 499D225×0035BS1
～	～ ～
C38	〃 〃

番号	抵抗
R1	カーボン3KΩ¼W VRD-ST2EY302J
R2	〃 1KΩ 〃 VRD-ST2EY102J
R3	〃 3KΩ 〃 VRD-ST2EY302J
～	～ ～ ～ ～
R10	〃 〃 〃 〃
R11	〃 1KΩ 〃 VRD-ST2EY102J
～	～ ～ ～ ～
R30	〃 〃 〃 〃
R31	金属皮膜 RN1 7.5ΩK 1W

番号	その他
U45	3端子レギュレータ MC7905CT
DP1	8回路DIPタイプスイッチDSS-108
DP2	〃
DP3	〃
P1	チエック端子 WP－2型A
～	～ ～
P10	〃 〃

LH8H03 ROM/RAMボード
アドレス・デコーダ部
(1/3)
R31以外の抵抗は1/4W

U30 74LS367
74LS32
U31
U40
U32 74LS02
U33 74LS00
U17 8T26
U18 8T26
*MREQ
*RFSH
*RD
*WR
*MS1
*MS2
*MS3
*MS4
*D0
*D1
*D2
*D3
*D4
*D5
*D6
*D7
*MREQi
RFSHi
RFSH + RD
*WRi
ROM0～ROM3　LH2708 又は i2716 用 ソケット
ROM4 ～ ROM7　LH2708 又は i2716 用ソケット
(注)（　）内は
2716の場合
ROM0
U1
ROM4
U5
ROM6
U2
ROM5
U6
ROM 2
U3
ROM 6
U7
ROM 3
U4
ROM 7
U8
Di0
Di1
Di2
Di3
Di4
Di5
Di6
Di7
*ROM0
*ROM1
*ROM2
*ROM3
*ROM4
*ROM5
*ROM6
*ROM7
*RENB
80T
(2/3)
+12V
-12V
-5V
U45
MC7905C
75452BP
U19
R31
7.5Ω 1W
R2
1K
C6
2.2 μF
LH8H03
データ・バッファ.　ROMブロック

●シャープ(株)は，Z80ファミリーについてZilog社と技術提携し，日本における実施権を保有しております．

●シャープ(株)は，Z80ファミリーに関するZilog社の刊行物の複製をする権利を同社から許諾されております．
読者は，本書のどの部分でもシャープ(株)に無断で複製したり，転載したり，または引用することはできません．

●本書は，Z80ファミリーを読者が正しく理解して戴くための教育手引書として編集しました．

●Z80ファミリーについてのデータとか，最新情報は，下記にお問合せ下さい．

シャープ株式会社

本　　　　社	〒545	大阪市阿倍野区長池町22番22号	☎ (06)621－1221（大代表）
電子部品事業本部	〒632	奈良県天理市櫟本町2613番地の1	☎ (07436)5-1321（大代表）
営業本部	〒545	大阪市阿倍野区長池町22番22号	☎ (06)621－1221（大代表）
東部地区営業	〒162	東京都新宿区市谷八幡町8番地	☎ (03)260－1161（大代表）
中部地区営業	〒460	名古屋市中区葵1丁目20番23号	☎ (052)931－5406、5424
西部地区営業	〒545	大阪市阿倍野区長池町22番22号	☎ (06)621－1221（大代表）

マイクロコンピュータZ80
ユーザーズマニュアル〔I〕©　　**定価 3,000円**

1981年9月1日　第1版第3刷発行

発　行／シャープ株式会社
発　売／株式会社エレクトロニクスダイジェスト書店部
〒105　東京都港区芝公園3-5-8 機械振興会館ビル(B3)
電　話　(03)434－3360(直)434－8211(代)

マイコンセミナー案内

●講師　梅原秀之
日本大学理工学部講師
(株)創英企画

●場所：機械振興会館セミナールーム(東京タワー直前) ☞館内は受付に掲示。時間：9:30〜16:30　●定員：15名
●事務局：東京都港区芝公園3-5-8 機械振興会館ビルB3F　エレクトロニクスダイジェスト　☎03-434-3360

コース	PART・1	PART・2	PART・3	PART・4	PART・5	PART・6	PART・7	PART・8	PART・9	PART・10
57スケジュール	2月12・13日 7月9・10日	2月26・27日 7月23・24日	'56 9月11・12日 3月12・13日 9月10・11日	9月25・26日 3月26・27日 9月24・25日	10月23・24日 4月9・10日 10月8・9日	10月30・31日 4月23・24日 10月22・23日	11月13・14日 5月7・8日 11月12・13日	11月27・28日 5月21・22日 11月26・27日	12月11・12日 6月11・12日 12月10・11日	12月25・26日 6月25・26日 12月24・25日
受講料	¥25,000	¥25,000	¥25,000	¥25,000	¥25,000	¥25,000	¥25,000	¥25,000	¥25,000	¥25,000

●特別受講料 1コース(同一企業の場合)、(2名　¥48,000)、(3名　¥69,000)、(4名　¥88,000)、(5名　¥105,000)

●開講のことば

マイクロコンピュータは，電子技術者の哲学を数㎜のシリコンチップ上に凝集させたものである．この哲学と発展過程を知ることなしにマイクロコンピュータを理解することは難かしい．本シリーズのセミナーはマイクロコンピュータ応用機器の開発，設計，製造，検査，保守等を目ざす技術者，マイクロコンピュータ応用機器の発注，使用する技術者を対象としており，全10コースから成り，全コース通し参加，独立コース参加の選択ができる，各2日間セミナーである．

日本でのマイクロコンピュータ導入時期からマイクロコンピュータ関連セミナーを広い分野にわたり，深くつっこんだ内容で永年開催して来たエレクトロニクスダイジェスト社が，過去の実績や参加者のアンケートをもとにより実戦的で即役に立つようにまとめたものである．各コースとも内容の重複をさけ，無駄のないよう全コースのカリキュラムが体系化されているため，参加者にとって理解しやすいセミナーである．

永年マイクロコンピュータ応用機器の開発を実地に手がけて来た講師によって講義や実習が行なわれ，また質疑応答も実地に即した内容で行なうことを旨としており，これも本セミナーの特長となっている．

1 マイクロコンピュータの完全理解コース

本コースは，初心者にもわかる平易な解説により，マイクロコンピュータの構成要素，基本構造と信号の流れ，動作順序とタイミング，マイコンの動作と命令語の関連，入/出力装置の接続と動作，入出力装置と割込，電源オンからプログラム実行までの手順，マイクロコンピュータ8080/8085/Z-80の概要を解説．

2 プログラミング・コース

マイクロコンピュータを動かすためのプログラムの作成手順と方法，アセンブル，シミュレーション/デバック，マイクロコンピュータの命令セットの解説と，基本的なI/O機器群が接続されたシステムに於て，それらインターフェイス回路の構成方法と，どのような手順によって，なぜI/O機器がソフトウェアにより動作するのか明解な講義

3 応用システム設計・コース

実用に近いマイクロコンピュータ応用機器例として1ボードCPUによる温度制御システムをとりあげ，システム開発・設計に伴うシステム分析，システム設計，開発ツールによるソフトウエア設計/デバッグ，P-ROM書込・とシステムの動作評価までの実際の手順，方法，演習問題，注意事項，トラブルシュート等について解説．

4 ワーク・ショップ・コース

参加者が理解しやすいI/O機器群を接続したシステムを，参加者自身が作成したプログラムで動かす実習をする．更に温度制御システム用プログラムを作成，シミュレーション/デバッグ，P-ROM書込及びシステムの動作評価までの実習を行なって，マイクロコンピュータ応用機器の開発，設計の手順・方法・道具立てなどを体得する．

5 Z80と周辺用LSI

Z-80CPUの構造と特長，レジスタ，命令，割込制御，ピン接続と使用法，周辺LSIの機能，動作，デイジー・チエーン優先割込制御使用法と制御プログラム，ダイナミックRAMのリフレッシュとそのインターフェース方法，注意事項など，Z-80CPU及び周辺用LSIを応用するのに必要な事柄をていねいに解説．

6 マイクロ・コンピュータのハードウェア設計

マイコンを構成するためによく利用されるTTLやICの解説，CPU及び制御回路，(8080, 8085, Z-80)，メモリ(ROM，スタチックRAM，ダイナミックRAM)，代表的なI/O機器インターフェースと制御プログラムの関連と，夫々のハードウェア設計に於ける問題点，注意事項とキーポイントについての解説．

7 I/O機器のインターフェースと制御プログラム(I)

I/O機器の割込と制御方法，マイクロコンピュータ用I/O機器(TTY，キーボード，表示器，アナログ・スイッチ，サンプルホルダ，A/Dコンバータ，D/Aコンバータ)とZ-80インターフェース用LSIによるインターフェース回路及びZ-80システムで実地に使用できるようサブルーチン化したOSを各々のI/O機器について完全解説．

8 I/O機器のインターフェースと制御プログラム(II)

DMAの基本と制御方法，マイクロコンピュータに接続されるあらゆるI/O機器(XYプロッタ，磁気ドラム，カセット磁気テープ，フロッピディスク)の解説とZ-80CPUとのインターフェース回路及びZ-80システムで実地に使用できるよう，サブルーチン化したOSを夫々のI/O機器について完全解説．

9 各種センサ・アクチェータとのインターフェースと制御

マイクロコンピュータの応用として，電子機器の分野はもとより，化学，医学，理学，機械，建築など広いこれらの分野で使用される代表的なセンサやアクチエータとマイクロコンピュータとのインターフェースの基本と制御プログラム及び実際のインターフェースの際の諸問題点と対策を実例に沿って解説．

10 開発機器の実例とドキュメント類の作成，まとめ方

開発された，マイクロコンピュータ応用機器，システムの実例を数機種とり上げ，発注者側の仕様書，開発者側との仕様決定方法，開発者側での承認仕様書，システム設計書，ハードウェア設計書，ソフトウェア設計書，取扱説明書，保守・メンテナンス説明書などドキュメント類の書き方整理の仕方まとめ方など．

●8080/8085/Z80の正しい使い方はPART・1〜5で終了します。●PART・4ワークショップコースはPART・3受講の方を受付ます。
●PART・5〜10コースは応用設計コースです。個々の選択ができますからテーマによって受講ください。
●このセミナーは年間数回、同一テーマで実施しております。日時・内容・その他の詳細は問い合せください。　エレクトロニクスダイジェスト (03)434-3360